

# Design and Analysis of A Majority Logic Based Imprecise 6-2 Compressor for Approximate Multipliers

Yongqiang Zhang, *Member, IEEE*, Cong He, Tingting Zhang, Jie Han, *Senior Member, IEEE*, Guangjun Xie

**Abstract**—Approximate computing is an emerging paradigm for trading off computing accuracy to reduce energy consumption and design complexity in a variety of applications, for which exact computation is not a critical requirement. Different from conventional designs using AND-OR and XOR gates, the majority gate is widely used in many emerging nanotechnologies. An ultra-efficient 6-2 compressor is proposed in this paper. It is composed of two majority gates that lead to low energy consumption and high hardware efficiency. The proposed compressor is utilized in the approximate partial product reduction of a modified 8×8 Dadda multiplier with a truncated structure. Experimental results show that this multiplier realizes a significant reduction in hardware cost, especially in terms of power and area, on average by up to 40% and 31% respectively, compared to exact and state-of-the-art designs. The application of image multiplication is also presented to assess the practicability of the multiplier. The results show that the proposed multiplier results in images with higher quality in peak signal to noise ratio (PSNR) and mean structural similarity index metric (MSSIM) compared to other designs.

**Keywords**—Approximate computing, majority logic, compressor, multiplier, image multiplication

## I. INTRODUCTION

As CMOS devices reach their fundamental physical limits, new nanoscale devices are considered for applications in digital circuit design with faster speeds, lower power consumption, and better scaling characteristics. Most of these nanotechnologies perform computation based on majority logic, such as magnetic tunnel junction (MTJ) devices.

Approximate computing has been widely used to save energy consumption of digital circuits for various error-tolerant applications, such as machine learning and multimedia processing, which allow a reasonable computational accuracy loss [1]. Multipliers are the most commonly used arithmetic units in these applications. Therefore, the speed and hardware cost of a multiplier has a significant impact on the overall performance of an application [2].

A multiplier design usually consists of three stages: 1) partial product generation, 2) partial product reduction, and 3) the computation of the final result [3]. Among these three stages, the second stage accounts for most of the power consumption and design complexity. Therefore, reducing the complexity of this step will significantly improve the efficiency of a multiplier.

In recent years, various approximate compressors have been designed for implementing energy-efficient multipliers. In [4], an approximate 4-2 compressor utilizes a majority gate, ignoring an input signal and assuming the output  $S$  equal to '1', which achieves improvements in terms of delay and power consumption compared to the previous designs. The approximate 4-2 compressor in [5] simplifies the circuit at a low accuracy loss. In [6], a 5-2 compressor ignores the carry-in and carry-out signals for hardware efficiency. Two approximate 5-2 compressors constructed by OR and AND gates have been proposed in [7], which drastically reduce the circuit complexity and power consumption.

In this paper, an ultra-efficient approximate 6-2 compressor based on majority logic is proposed and an energy-efficient approximate multiplier based on the approximate partial product reduction using the proposed approximate compressor is designed. The main contributions of this work are summarized as follows: (1) An approximate 6-2 compressor design by using two majority gates for efficient compression, (2) An 8×8 approximate truncated Dadda multiplier using the proposed compressor with a significant reduction in power consumption, delay, and area compared with existing designs, and (3) Image multiplication to illustrate the use of the proposed multiplier in practical applications.

The rest of the paper is organized as follows. Section II discusses the proposed compressor and multiplier. The simulation results and the application of image multiplication are presented in Section III. Section IV gives the conclusion.

## II. THE PROPOSED COMPRESSOR AND MULTIPLIER

### A. The Exact 6-2 Compressor

As shown in Fig. 1, a precise 6-2 compressor is composed of two full adders and one 4-2 compressor [8]. It has six main inputs ( $x_1, x_2, x_3, x_4, x_5, x_6$ ), three input carry ( $cin_1, cin_2, cin_3$ ), two main outputs ( $sum$  and  $carry$ ), and three carry outputs ( $cout_1, cout_2, cout_3$ ). Its logical expression is

$$\begin{aligned} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + cin_1 + cin_2 + cin_3 \\ & = sum + 2 \times (carry + cout_1 + cout_2 + cout_3) \end{aligned} \quad (1)$$

The output  $sum$  has the same significance with the inputs, while the other four outputs have a higher significance, which propagate in the computation process as higher significant bits.

This work was supported by Fundamental Research Funds for the Central Universities of China (Grant No. JZ2020HGTA0085, No. JZ2020HGQA0162), by the Natural Sciences and Engineering Research Council (NSERC) of Canada (Project Number: RES0048688).

Y. Zhang, C. He, and G. Xie are with the School of Microelectronics, Hefei University of Technology, Hefei 230009, China (e-mail: ahzhangyq@hfut.edu.cn; 2191158315@qq.com; gjxie8005@hfut.edu.cn)

T. Zhang and J. Han are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: ttzhang@ualberta.ca, jhan8@ualberta.ca)

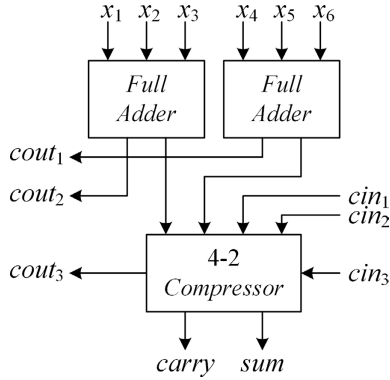


Figure 1. The conventional 6-2 compressor [8].

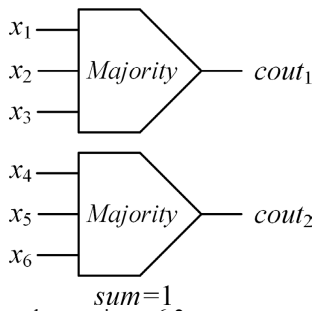


Figure 2. The proposed approximate 6-2 compressor.

### B. The Proposed Approximate 6-2 Compressor

As shown in Fig. 2, an approximate 6-2 compressor is proposed to implement the functions in (2), (3), and (4):

$$cout_1 = \text{Majority}(x_1, x_2, x_3) = x_1(x_2 + x_3) + x_2x_3. \quad (2)$$

$$cout_2 = \text{Majority}(x_4, x_5, x_6) = x_4(x_5 + x_6) + x_5x_6. \quad (3)$$

$$sum = 1. \quad (4)$$

It can be seen that the compressor is composed of two majority gates, where  $cin$ ,  $cout_3$ , and  $carry$  are ignored. Since the output  $sum$  is always equal to '1', there is no additional hardware for computing it, which significantly reduces the power and delay of the design. The approximate function of the compressor is described in

$$\begin{aligned} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ & = 2 \times (cout_1 + cout_2) + sum \end{aligned} \quad (5)$$

TABLE I shows the truth table of the proposed 6-2 compressor. The error distance (ED) is defined as the arithmetic distance between the approximate and exact values [9]. As one of the principal error metrics for designing an approximate compressor, the value of ED should be as small as possible. According to TABLE I, the proposed compressor produces incorrect results in 32 cases out of 64 input combinations. None of the ED values exceeds 1, so it avoids unacceptably large errors when applying the proposed compressor in multiplier designs. Moreover, the opposite value of ED such as -1 and 1 can eliminate the impact of each other in a multiplier [4].

TABLE I. THE TRUTH TABLE AND ERROR DISTANCE OF THE PROPOSED 6-2 COMPRESSOR

$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$cout_1$	$cout_2$	Error distance
0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	1	1	0	1	1
0	0	0	1	0	0	0	0	0
0	0	0	1	0	1	0	1	1
0	0	0	1	1	0	0	1	1
0	0	0	1	1	1	0	1	0
0	0	1	0	0	0	0	0	0
0	0	1	0	0	1	0	0	-1
0	0	1	0	1	0	0	0	-1
0	0	1	0	1	1	0	1	0
0	0	1	1	0	0	0	0	-1
0	0	1	1	0	1	0	0	-1
0	0	1	1	1	0	0	1	0
0	0	1	1	1	1	0	1	-1
0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	-1
0	1	0	0	1	0	0	0	-1
0	1	0	0	1	1	0	1	0
0	1	0	1	0	0	0	0	-1
0	1	0	1	0	1	0	0	0
0	1	0	1	1	0	0	1	0
0	1	0	1	1	1	0	1	-1
0	1	1	0	0	0	1	0	0
0	1	1	0	0	1	0	0	0
0	1	1	0	1	0	0	1	0
0	1	1	0	1	1	0	1	-1
0	1	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0	0
0	1	1	1	1	0	1	1	1
0	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
1	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	-1
1	0	0	0	1	0	0	0	-1
1	0	0	0	1	1	0	1	0
1	0	0	1	0	0	0	0	-1
1	0	0	1	0	1	0	1	0
1	0	0	1	1	0	0	1	-1
1	0	0	1	1	1	0	1	-1
1	0	1	0	0	0	1	0	1
1	0	1	0	0	1	0	0	0
1	0	1	0	1	0	0	1	0
1	0	1	0	1	1	0	1	0
1	0	1	1	0	0	1	0	0
1	0	1	1	0	1	0	0	0
1	0	1	1	1	0	0	1	0
1	0	1	1	1	1	0	1	-1
1	0	1	1	1	1	1	0	0
1	1	0	0	0	0	1	0	1
1	1	0	0	0	1	0	0	0
1	1	0	0	1	0	1	0	0
1	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	0
1	1	0	1	0	0	1	0	0
1	1	0	1	0	1	0	0	-1
1	1	0	1	1	0	1	0	-1
1	1	0	1	1	1	1	1	0
1	1	0	1	1	1	1	1	0
1	1	1	0	0	0	1	0	0
1	1	1	0	0	1	0	0	-1
1	1	1	0	1	0	1	0	-1
1	1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1	0
1	1	1	1	0	0	1	0	-1
1	1	1	1	0	1	0	0	-1
1	1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1	-1

### C. The Approximate Multiplier

The 8×8 Dadda multiplier is considered to evaluate the efficacy of the proposed compressor in approximate

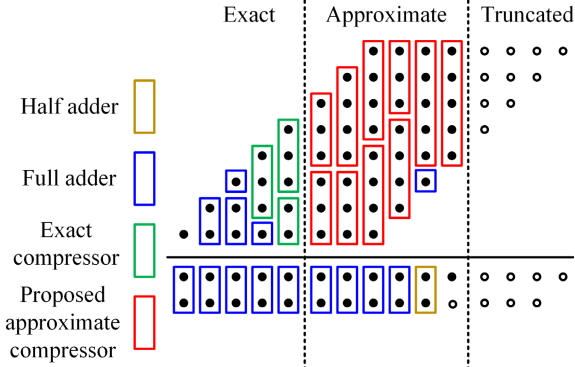


Figure 3. Partial product reduction of the proposed multiplier.

TABLE II. HARDWARE COST OF 6-2 COMPRESSORS

Designs	Area ( $\mu\text{m}^2$ )	Power ( $\text{mW}$ )	Delay ( $\text{ns}$ )	PDP ( $\text{fJ}$ )	EDP ( $\text{fJ}\times\text{ns}$ )
Proposed	9.00	$1.11\times 10^{-3}$	0.25	0.28	0.07
Exact	36.00	$8.64\times 10^{-3}$	0.58	5.01	2.91

multipliers [10]. First, an array of partial products is generated by using AND gates. These products are then compressed by using compressors. Finally, the final results are computed by a ripple carry adder.

The proposed approximate reduction process for a Dadda multiplier is shown in Fig. 3, where each dot represents a partial product bit. The four least significant columns are truncated for further hardware reduction, which has a negligible impact on the accuracy of multipliers. The approximate compressor is only used in the middle six columns, whereas the five most significant columns use exact compressors to guarantee accuracy.

In the first step, as shown in Fig. 3, the partial products are compressed into two rows by using ten approximate 6-2 compressors, five full adders, and three exact compressors. Then, a half adder and nine full adders are used to compute the final results. Due to the high efficiency of the proposed compressor, only two compression steps are required to obtain the final result.

### III. SIMULATION RESULTS AND APPLICATION

In this section, the circuit designs are all described in Verilog HDL language and synthesized in Synopsys Design Compiler with a TSMC 65 nm standard cell library. The evaluation of accuracy and the application in image multiplication are implemented through MATLAB.

#### A. Approximate Compressors

Since there is no approximate 6-2 compressor in previous works, the proposed design is compared with the exact 6-2 compressor. TABLE II presents the comparison of the proposed compressor and the exact compressor in hardware metrics, including area, power, delay, power-delay product (PDP), and energy-delay product (EDP). According to TABLE II, the proposed design improves the power by 87%, the delay by 57%, and the area by 75% compared to the exact 6-2 compressor because of the extremely simple structure.

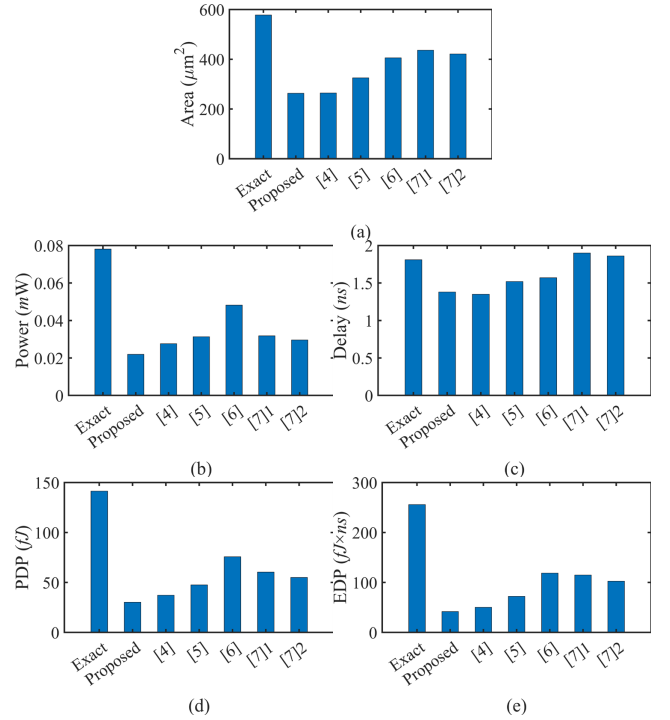


Figure 4. The comparison of approximate multipliers in hardware.

TABLE III. THE RESULTS IN ER, MED, AND NMED OF APPROXIMATE MULTIPLIERS

Multipliers	ER (%)	MED	NMED
Proposed	99.89	$7.89\times 10^2$	$12\times 10^{-3}$
[4]	99.82	$4.94\times 10^2$	$7.60\times 10^{-3}$
[5]	85.73	$2.24\times 10^3$	$34.41\times 10^{-3}$
[6]	100	$56.2\times 10^3$	$86.46\times 10^{-3}$
[7]1	100	$1.18\times 10^4$	$182.11\times 10^{-3}$
[7]2	99.99	$1.18\times 10^4$	$182.02\times 10^{-3}$

#### B. Approximate Multipliers

**Hardware cost:** The comparison of the proposed and existing approximate multipliers is shown in Fig. 4. The proposed multiplier has the lowest power, area, PDP, and EDP, due to the use of the ultra-efficient approximate 6-2 compressor and the simplified structure of the multiplier. It achieves an improvement in area and power by 31% and 40% on average, respectively.

**Computing accuracy:** The error rate (ER), the mean error distance (MED), and the normalized mean error distance (NMED) are considered to evaluate the accuracy of multipliers.

The ER is defined as the percentage of producing incorrect output. The MED is the average of the error distance, which is expressed as [11]

$$MED = \frac{1}{2^{2N}} \sum_{i=1}^{2^{2N}} |ED_i|, \quad (6)$$

where  $ED_i$  is the ED for the  $i^{\text{th}}$  input combination and  $N$  is the bit width of a multiplier.

The NMED is defined as the normalization of MED by the maximum output of the exact design, given by

TABLE IV. THE PSNR AND MSSIM OF MULTIPLIED IMAGES USING 8×8 MULTIPLIERS

Multipliers	PSNR (dB)		MSSIM	
	Clock× FemaleRGB	TreeRGB× HouseRGB	Clock× FemaleRGB	TreeRGB× HouseRGB
Proposed	38.80	34.78	0.9270	0.9835
[4]	41.36	37.91	0.9795	0.9858
[5]	27.29	28.71	0.8962	0.9257
[6]	23.05	20.97	0.8366	0.8223
[7]1	17.43	15.15	0.8317	0.7607
[7]2	17.38	15.17	0.8373	0.7589

$$NMED = \frac{1}{(2^N - 1)^2} \sum_{i=1}^{2^N} \frac{|ED_i|}{2^{2^N}}. \quad (7)$$

The results in accuracy metrics of the proposed and existing multipliers are compared in TABLE III. The truncated structure of the proposed design leads to a high error rate. The proposed multiplier has a lower value of MED and NMED than other designs except [4], while it leads to an acceptable quality in image processing, which is introduced in the next subsection.

### C. Image Multiplication

To assess the applicability of the approximate multipliers in fault-tolerant applications, they are applied in image multiplication by multiplying two images pixel by pixel. The images are downloaded from the USC-SIPI Image Database [12]. The PSNR and the *MSSIM* are applied to evaluate the quality of processed images.

The *PSNR* is given by

$$PSNR = 10 \log_{10} \left( \frac{w \times r \times MAX^2}{\sum_{i=0}^{w-1} \sum_{j=0}^{r-1} [S'(i, j) - S(i, j)]^2} \right), \quad (8)$$

where  $w$  and  $r$  represent the image dimensions,  $S'(i, j)$  and  $S(i, j)$  are the pixels in exact and approximate images, and  $MAX$  is the maximum pixel value in the image. A *PSNR* value higher than 30 dB is usually considered good enough [13].

The *MSSIM* is defined as

$$MSSIM(X, Y) = \frac{1}{k} \sum_{i=1}^k \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (9)$$

where  $X$  and  $Y$  are approximate and exact images, and  $x$  and  $y$  represent the window size of the approximate and the exact images. The explanations for other parameters can be found in [14] in detail. The closer the *MSSIM* value is to 1, the more similar the processed image will be to the reference one.

The *PSNR* and *MSSIM* values of two image multiplication examples are shown in TABLE IV. Both the *PSNR* values of the proposed design are larger than 30 dB. Moreover, the *MSSIM* values are close to 1. Therefore, the approximate multiplier has a great potential for image processing.

## IV. CONCLUSION

In this work, an ultra-efficient approximate 6-2 compressor has been proposed by utilizing two majority gates. The experimental results show that it achieves a significant improvement in area, power, and delay compared to the exact design. Moreover, the proposed approximate compressor is utilized in a specialized truncated Dadda multiplier. Evaluated from accuracy and hardware metrics, the proposed multiplier improves power by 40% and area by 31% on average with a decrease of 87% in NMED, compared to the latest approximate multiplier design. The case study of image multiplication shows the multiplier using the proposed approximate compression method can be successfully utilized in image processing applications with the purpose of achieving a design with low energy and acceptable accuracy.

## REFERENCES

- [1] G. Park, J. Kung, and Y. Lee, "Design and analysis of approximate compressors for balanced error accumulation in MAC operator," *IEEE Trans. Circuits Syst. I-Regul. Pap.*, vol. 68, no. 7, pp. 2950-2961, Jul 2021.
- [2] F. Salmanpour, M. Moaiyeri, and F. Sabetzadeh, "Ultra-compact imprecise 4:2 compressor and multiplier circuits for approximate computing in deep nanoscale," *Circuits Syst. Signal Process.*, vol. 40, no. 9, pp. 4633-4650, Sep 2021.
- [3] S. Angizi, H. Jiang, R. DeMara, J. Han, and D. Fan, "Majority-based spin-CMOS primitives for approximate computing," *IEEE Trans. Nanotechnol.*, vol. 17, no. 4, pp. 795-806, Jul 2018.
- [4] F. Sabetzadeh, M. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," *IEEE Trans. Circuits Syst. I-Regul. Pap.*, vol. 66, no. 11, pp. 4200-4208, Nov 2019.
- [5] S. Venkatachalam, and S. Ko, "Design of power and area efficient approximate multipliers," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 25, no. 5, pp. 1782-1786, May 2017.
- [6] M. Ahmadinejad, and M. Moaiyeri, "Energy- and quality-efficient approximate multipliers for neural network and image processing applications," *IEEE Trans. Emerging Top. Comput.*, pp. 1-1, Apr. 2021.
- [7] A. Gorantla, and P. Deepa, "Design of approximate compressors for multiplication," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 3, pp. Article 44, May 2017.
- [8] A. Rahnamaei, "CMOS high-performance 5-2 and 6-2 compressors for high-speed parallel multipliers," *Inf. Midem-J. Microelectron. Electron. Compon. Mater.*, vol. 50, no. 2, pp. 115-123, Jun 2020.
- [9] 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.
- [10] 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.
- [11] H. L. Jiang, C. Liu, L. B. Liu, F. Lombardi, and J. Han, "A review, classification, and comparative evaluation of approximate arithmetic circuits," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 4, pp. Article 60, Aug 2017.
- [12] "The usc-sipi image database," <https://sipi.usc.edu/database/>.
- [13] M. Ansari, H. Jiang, B. Cockburn, and J. Han, "Low-power approximate multipliers using encoded partial products and approximate compressors," *IEEE J. Emerging Sel. Top. Circuits Syst.*, vol. 8, no. 3, pp. 404-416, Sep 2018.
- [14] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-12, Apr 2004.