International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 4, December 2026, pp. 1624~1634
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i4.pp1624-1634

Approximate computing circuits for error-tolerant applications with low energy consumption

Anusha Gorantla[1], Vamsidhar Anagani[1], Venkata Sridhar Gummadidala[2], Ramesh Babu Geddam[1]
[1]Department of Electronics and Communication Engineering, Raghu Engineering College, Visakhapatnam, India
[2]Department of Electronics and Communication Engineering, Dadi Institute of Engineering and Technology, Visakhapatnam, India

Article Info

Article history:

Received Feb 28, 2025
Revised Jul 25, 2026
Accepted Aug 28, 2026

Keywords:

Approximate computing
Arithmetic circuits
Energy efficiency
Image processing applications
PSNR

ABSTRACT

Energy efficiency is a crucial factor in very large-scale integrated (VLSI) circuit design, and traditional methods offer limited solutions to power-intensive circuits. Recently, approximate computing techniques have emerged, enhancing key design parameters like power consumption, propagation delay, and area, albeit with a compromise in accuracy. These techniques are especially suitable for high-performance computing, as they offer reduced energy consumption while compromising some accuracy, making them ideal for use in arithmetic circuits. This paper introduces enhanced versions of approximate adders (AxAs) and subtractors (AxSCs) for fundamental operations. Building upon these designs, approximate multipliers using the Wallace tree multiplier (AxWTM) and approximate dividers (AxDs) have been developed, and their performances have been compared regarding power consumed, spatial requirements, delay time, and error frequency. Many digital image and video processing applications, along with other multimedia tools, can function effectively even with slight inaccuracies in their output results. The existing and proposed designs were applied to image squaring, image multiplication, image subtraction, and image division applications, and peak signal-to-noise ratio (PSNR) was obtained. It is found that the suggested approximate designs achieved a better PSNR for all image applications, and the results were superior to those of their counterparts.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Vamsidhar Anagani
Department of Electronics and Communication Engineering, Raghu Engineering College
Visakhapatnam, India
Email: vamsianagani@gmail.com

1. INTRODUCTION

Arithmetic circuits, such as adders and multipliers, are crucial components in many processing circuits, such as digital signal processors (DSPs) [1]. In contrast to adders, multipliers demand more hardware resources and take longer to function. In digital processing design, multipliers significantly influence the overall performance of the system and are employed to execute algorithms such as filtering, convolution, and arithmetic logic operations. Addition and multiplication are the most commonly used operations in very large-scale integration (VLSI) systems [2]. Because digital circuitry is a fundamental component of computer architectures, the use of approximate computing enables the reduction of logic complexity and the creation of low-cost circuit designs [3]. Approximate computing techniques enhance important design parameters, such as power consumption, propagation delay, and area, with a trade-off in accuracy. These techniques are especially suitable for high-performance computing, making them ideal for arithmetic circuits. Researchers have proposed various approximate adders (AxAs) and subtractors, which are

often cumbersome and require additional space for implementation [4]–[7]. These approximate circuits employ fewer gates in their architectures. This study focuses on developing designs that occupy less space than current and precise models. To maintain accuracy, approximate computing techniques have not been implemented for the most significant bits (MSBs) [8].

This study introduces enhanced versions of AxAs and approximate subtractors (AxSCs) for fundamental operations. Based on these enhanced versions of AxAs and AxSCs, approximate multipliers using the wallace tree multiplier (AxWTM) and approximate dividers (AxDs) were developed, and their performances were compared with those of existing approximate circuits in terms of power consumption, area occupied, delay time, and error count. The proposed designs were applied to image processing applications, such as squaring, multiplication, subtraction, and division of images. The peak signal-to-noise ratio (PSNR) was obtained for all the cases of existing and proposed approximate circuits, and it was observed that the proposed designs perform a better PSNR for all image applications.

The remainder of this paper is organized as follows: Section 2 discusses existing AxAs and Subtractors. Section 3 explains the construction of the proposed AxAs and Subtractors, which helps in designing the Approximate Multipliers and dividers. Section 4 presents the experimental results and demonstrates the implementation of the aforementioned approximate circuits in image processing. Finally, section 5 provides the conclusions.

2. EXISTING DESIGNS

The fundamental component of digital circuit design is a full adder. Approximate full adders have emerged as a new methodology for achieving approximate computing designs that are low in power consumption with an increase in speed [9]–[11]. Current AxAs incorporate errors, resulting in reduced energy consumption and minimal effects on output quality. These adders enhance performance at the cost of accuracy. Two approaches have been used in existing AxA designs [3]. To further improve performance, it is necessary to develop various new AxAs. Similarly, existing AxSC designs employ two approaches [12]. Additional new AxSCs must be developed to further improve performance. The proposed AxAs and AxSCs can be used to design approximate AxWTMs and AxDs.

2.1. Approximate adders (AxAs)

Current AxA designs employ two strategies: (1) approximating only the sum and (2) approximating both the sum and C_{out}. The corresponding logical expressions for these approaches can be found in [13]. This study examined several existing AxAs that offer improved design metrics while maintaining a reasonable accuracy. Table 1 lists the sum and C_{out} values of the adders. Among the existing designs, AxA5, AxA6, and AxA1 [3] used the second approach, resulting in six, five, and three errors, respectively. These errors stemmed from the various input combinations. Approximating the C_{out} output introduces carry errors that propagate through multiple stages, thereby amplifying the overall-design error. By approximating both the sum and C_{out} outputs, these designs achieve enhanced power efficiency and speed, albeit with increased errors. By contrast, AxA10 [3] employs the first approach (approximating only the sum), producing three errors and delivering moderate design metrics with acceptable error rates (ER).

Table 1. Truth table for existing AxAs [3]

Inputs			ExFA		AxA5		AxA6		AxA1		AxA10	
A	*B*	C_{in}	*Sum*	C_{out}	*Sum*	C_{out}	*Sum*	C_{out}	*Sum*	C_{out}	*Sum*	C_{out}
0	0	0	0	0	0	0	1×	0	0	0	1×	0
0	0	1	1	0	0×	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	0×	1×	1	0
0	1	1	0	1	1×	0×	1×	0×	0	1	1×	1
1	0	0	1	0	0×	1×	0×	1×	0×	0	0×	0
1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	0	0	1	1×	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1

Note: × denotes an error in the bit position

2.2. Approximate subtractors (AxSCs)

Current AxSC designs focus only on approximating the difference (D). Table 2 illustrates a comparison of the output borrow (B_{out}) and the differences (D) between the exact subtractor (ExSC) and AxSCs. In all existing subtractor designs, the errors are limited to the difference (D) bit. The corresponding logic equations can be found in [12].

Table 2. Truth table for existing AxSCs [12]

Inputs			ExSC		AxSC4		AxSC5		AxSC6		AxSC7	
X	*Y*	B_{in}	*D*	B_{out}	*D*	B_{out}	*D*	B_{out}	*D*	B_{out}	*D*	B_{out}
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1	0×	1
0	1	1	0	1	1×	1	0	1	0	1	0	1
1	0	0	1	0	1	0	1	0	0×	0	1	0
1	0	1	0	0	0	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1

Note: × denotes an error in the bit position

3. PROPOSED DESIGNS

3.1. Proposed AxAs

For addition operations, the precision of C_{out} surpassed that of Sum. Both the suggested and existing AxAs employ two distinct approaches. Table 3 presents the sum and C_{out} values for the proposed adders. The proposed AxAs exhibits error bits in both the sum and C_{out}, whereas AxA10 displays error bits exclusively in the sum.

Proposed AxA1: The design of the Proposed AxA1 employs approximations for both the sum and C_{out} methods. This approach resulted in three and two errors in the sum and C_{out}, respectively (Table 3). K-map simplification was used to derive logic expressions (1) and (2) for the proposed AxA1.

$$\text{Sum} = A'(B' + C'_{in}) \; + \; A'B'C'_{in} \tag{1}$$

$$\mathrm{C_{out}} = A \tag{2}$$

Proposed AxA2: In the Proposed AxA2 design, an approximation is utilized for the sum approach, resulting in four errors for the sum, while leaving C_{out} unaffected. The proposed AxA2 logical expressions are defined as (3), (4):

$$\text{Sum} = B \tag{3}$$

$$\mathrm{C_{out}} = AB \; + \; BC_{in} \; + \; AC_{in} \tag{4}$$

Table 3. Truth table for proposed AxAs

Inputs			ExFA		AxA1		AxA2	
A	*B*	C_{in}	*Sum*	C_{out}	*Sum*	C_{out}	*Sum*	C_{out}
0	0	0	0	0	1×	0	0	0
0	0	1	1	0	1	0	0×	0
0	1	0	1	0	1	0	1	0
0	1	1	0	1	0	0×	1×	1
1	0	0	1	0	0×	1×	0×	0
1	0	1	0	1	0	1	0	1
1	1	0	0	1	0	1	1×	1
1	1	1	1	1	0×	1	1	1

Note: × denotes an error in the bit position

3.2. Proposed AxSCs

The proposed AxSCs offer the potential for additional logic optimization and enhanced performance [14]. Table 4 displays the truth table for the suggested designs of AxSCs.

Proposed AxSC8: The APSC8 design incorporates an approximation only in the difference (D), resulting in a single error. The APSC8's logical expressions are defined as (5), (6):

$$\mathrm{D} = X'YB_{in} + XY'B'_{in} + XYB_{in} \tag{5}$$

$$\mathrm{B_{out}} = (X \oplus Y)'B_{in} + X'Y \tag{6}$$

Proposed AxSC9: The AxSC9 design uses approximation methods for both the D and B_{out} approaches, resulting in two and one errors in D and B_{out}, respectively. The logical expressions for APSC9 are as (7), (8):

$$D = B_{in} + X'Y + XY' \quad (7)$$

$$B_{out} = X' + YB_{in} \quad (8)$$

Proposed AxSC10: The AxSC10 design incorporates approximations in both the D and B_{out} approaches, resulting in four errors. As outlined in Table 4, two of these errors were associated with D, whereas the other two were specific to B_{out}. The logic (9) and (10) of AxSC10 are shown,

$$D = B_{in} + X'Y + XY' \quad (9)$$

$$B_{out} = X' \quad (10)$$

Table 4. Truth table for proposed AxSCs

Inputs			ExSC		AxSC8		AxSC9		AxSC10	
X	*Y*	B_{in}	*D*	B_{out}	*D*	B_{out}	*D*	B_{out}	*D*	B_{out}
0	0	0	0	0	0	0	0	1×	0	1×
0	0	1	1	1	0×	1	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1
0	1	1	0	1	0	1	1×	1	1×	1
1	0	0	1	0	1	0	1	0	1	0
1	0	1	0	0	0	0	1×	0	1×	0
1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0×

Note: × denotes an error in the bit position

3.3. Approximate wallace tree multiplier (AxWTM) using proposed AxAs

The design of energy-efficient digital systems requires rapid multiplication. However, multipliers consume substantial hardware resources and processing time [15]. The complexity of designing multipliers increases with the requirement for multiple adders, resulting in irregular structures, increased power usage, and design challenges. The current research aims to replace exact multipliers with approximate alternatives [16], [17]. This section introduces various AxWTMs based on the proposed AxAs. Designs utilizing the proposed AxA1 and AxA2 were designated AxWTM1 and AxWTM2, respectively [18]. In Figure 1, least significant bits (LSBs) are denoted as F_{00}-F_{02}, F_{06}-F_{08}, F_{10}-F_{12}, F_{16}-F_{18}, F_{20}-F_{22}, F_{30}-F_{33}, while MSBs are labeled F_{03}-F_{05}, F_{09}-F_{11}, F_{13}-F_{15}, F_{19}-F_{21}, F_{23}-F_{25}, F_{34}-F_{36}. To achieve superior performance compared with conventional 8×8 WTM, the proposed AxAs were applied to LSBs, and exact full adders (ExFAs) were used for MSBs [19].

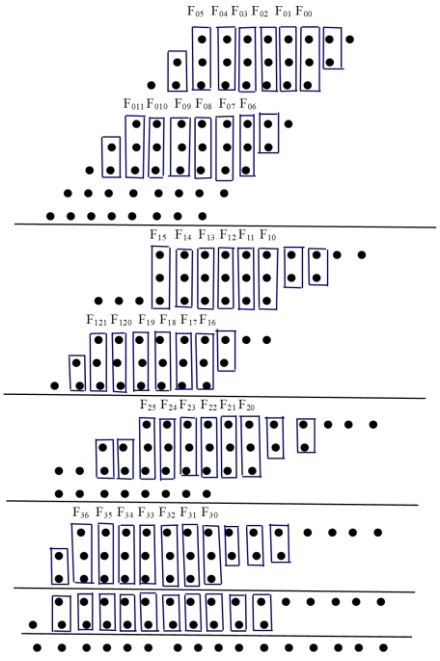


Figure 1. 8×8 Approximate wallace tree multipler (AxWTM) using proposed AxAs

3.4. Approximate divider (AxD) using proposed AxSCs

As the construction of multipliers uses adders, in a similar manner, the implementation of dividers requires the subtractors. The restoring AxD utilizes a controlled subtractor (CS) block, as described in [20]. Various AxDs have been designed by incorporating existing and proposed AxSCs into the CS block. Figure 2 illustrates an 8×8 AxD constructed using Existing and AxSCs [21]. The 8-bit input X [7:0] is divided by another 8-bit input Y [7:0], producing a 6-bit quotient Q [5:0] and a 4-bit remainder R [3:0]. Similar to the AxWTM approach, the proposed AxSCs were applied to LSBs, and ExSCs were used for MSBs to achieve superior performance compared with the conventional 8×8 Array Divider. The remainder R is obtained from AxSC at the LSB level, whereas the quotient Q is deduced from ExSCs at the MSB level with the help of an OR gate [22].

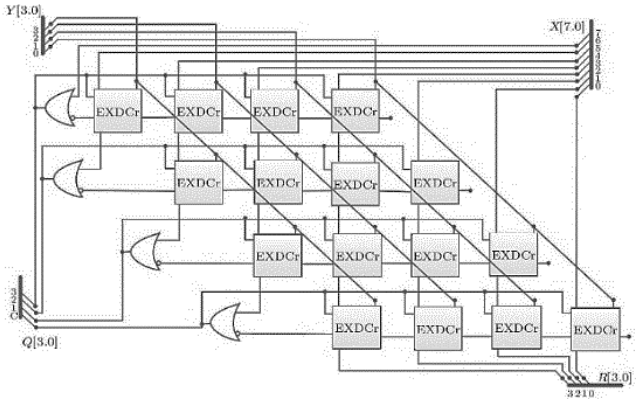


Figure 2. AxD using proposed AxSCs

4. RESULTS AND DISCUSSION

The proposed AxAs, AxSCs, AxWTMs, and AXDs detailed in the previous section were simulated, and the results are analyzed in this section. Each design was developed in Verilog and synthesized with the help of a synopsis design compiler, utilizing 45 nm technology. The power consumption in nanowatts (nW), time delay in picoseconds (ps), area (spatial) requirements in square micrometers (μm^2), and several errors of the proposed designs were measured and compared with those of existing approximate designs. Subsequently, both the existing and proposed designs were utilized in various image-processing techniques, including squaring, multiplication, subtraction, and division of images. These imaging methods were evaluated using the PSNR.

4.1. Approximate adders (AxAs)

Table 5 presents the comparative analysis of power consumption, propgation delay, area occupied, along with error count among the ExFAs, existing AxAs, and proposed AxAs for 45 nm technology. The current AxAs AxA5, AxA6, AxA1, and AxA10 exhibited six, five, three, and three errors, respectively. In contrast, the newly proposed AxA1 displayed three sum errors and two C_{out} errors, whereas AxA2 exhibited four errors exclusively in the sum. ExFA consumed the highest power at 595Nw, whereas AxA5, AxA6, AxA1, and AxA10 used 38nW, 78nW, 315nW, and 118nW, respectively. The proposed AxA1 and AxA2 consumed 45nW and 30nW, respectively.

Table 5. Comparitive analysis of AxAs

AxAs	Power (nW)	Delay (ps)	Area (μm^2)	Number of errors
ExFA	595	74	13	0
Existing AxAs [3]				
AxA5	38	16	7	6
AxA6	78	23	5	5
AxA1	315	56	13	3
AxA10	118	32	10	3
Proposed AxAs				
AxA1	45	23	6	5
AxA2	30	16	5	4

Among all the AxAs, both existing and proposed, AxA2 demonstrated superior performance in terms of power consumption. Additionally, AxA2 surpassed all existing AxAs in terms of delay time, with a

minimum value of 16ps. Regarding spatial requirements, both the existing AxA6 and proposed AxA2 occupied 5μm^2. Considering all the evaluated metrics, the proposed AxA2 outperformed all existing AxAs.

4.2. Approximate subtractors (AxSCs)

Table 6 illustrates the performance metrics of the ExSC and the existing and newly proposed AxSCs using 45 nm technology. These metrics include power usage, time delay, area requirements, and error frequency. ExSC, AxSC4, AxSC5, AxSC6, and AxSC7 each had a single error, whereas the newly introduced AxSC8, AxSC9, and AxSC10 showed one, three, and four errors, respectively. Among the new designs, AxSC10 consumed the least power at hiding 215nW, whereas AxSC8 and AxSC9 used 429nW and 395nW, respectively. AxSC10 also demonstrated the quickest response time of 49ps, compared with 78ps for AxSC9 and 110ps for AxSC8. The power consumption of AxSC10 was 51% lower than that of AxSC4, primarily because fewer logic components were used. Furthermore, AxSC10 achieved a 49% decrease in the delay compared with AxSC6. Among the three new AxSCs presented, AxSC10 distinguished itself by significantly reducing power consumption, response time, and spatial requirements.

Table 6. Comparitive analysis of AxSCs

AxSCs	Power (nW)	Delay (ps)	Area (μm^2)	Number of errors
ExSC	1410	96	15	0
Existing AxSCs [22]				
AxSC4	420	96	10	1
AxSC5	595	91	13	1
AxSC6	692	88	19	1
AxSC7	723	98	15	1
Proposed AxAs				
AxSC8	429	110	9	1
AxSC9	395	78	5	3
AxSC10	215	49	3	4

4.3. Approximate multipliers (AxWTMs)

Table 7 presents a comparative analysis of different 8×8 multiplier designs, including the traditional WTM and both existing and proposed AxWTMs that employ various AxAs. The comparison evaluates these designs based on power usage, processing speed, spatial requirements, and error frequency. The AxWTMs utilizing the existing AxA5, AxA6, AxA1, and AxA10 exhibited 90, 75, 45, and 45 errors, respectively, whereas the AxWTMs employing the newly proposed AxA1 and AxA2 demonstrated 75 and 60 errors, respectively. The precise 8×8 WTM consumed the most power at 12351nW, whereas AxWTMs using existing AxA5, AxA6, AxA1, and AxA10 used 10174nW, 11027nW, 10863nW, and 11976nW, respectively. The AxWTMs using the newly proposed AxA1 and AxA2 consumed 8952nW and 9836nW, respectively. Among all the current and proposed AxWTMs, AxA1 showed a minor decrease in power consumption. Considering the additional two metrics, AxWTM employing the suggested AxA1 outperformed all alternative designs in terms of overall effectiveness.

Table 7. Comparitive analysis of AxWTMs

AxWTMs	Power (nW)	Delay (ps)	Area (μm^2)	Number of errors
8×8 WTM [13]	12351	946	464	0
Existing AxWTMs [3]				
AxA5	10174	435	219	90
AxA6	11027	678	364	75
AxA1	10863	1069	342	45
AxA10	11976	992	318	45
Proposed AxAs				
AxWTM1	8952	207	89	75
AxWTM2	9836	589	412	60

4.4. Approximate dividers (AxDs)

The performance metrics of the ExD and AxDs based on the existing and proposed AxSCs are listed in Table 8. Among the new designs, AxD10 consumed the least power at 20949nW, whereas AxD8 and AxD9 consumed 52683nW and 36690nW, respectively. AxD10 also demonstrated the quickest response time of 3458ps, compared to 6995ps for AxD8 and 5734ps for AxSC9. The power consumption of AxD10 was 60% lower than that of AxD4, primarily because fewer logic components were used. Given that this study prioritizes approximation accuracy, the proposed AxD10 demonstrates superior performance to existing AxDs.

Table 8. Comparitive analysis of AxDs

AxDs	Power (nW)	Delay (ps)	Area (μm²)	Number of errors
ExD	57849	7006	1849	0
		Existing AxDs [14]		
AxD4	34903	7047	1516	10
AxD5	39532	6269	1503	10
AxD6	43788	6801	1683	10
AxD7	38174	6611	1504	10
		Proposed AxDs		
AxD8	52683	6995	1690	10
AxD9	36690	5734	1444	30
AxD10	20949	3458	955	40

4.5. Accuracy evaluation

Various error metrics were utilized to evaluate how the proposed AxAs and AxSCs perform compared to existing designs. These included the error distance (ED), ER, pass rate (PR), mean ED (MED), and normalized ED (NED) [23]. The ED represents the absolute difference between the approximate and exact computation values, whereas the MED was calculated as the average of all EDs. The definitions of ER and PR are as (11), (12):

$$\mathrm{ER} = \frac{|\text{Approximate Value} - \text{Exact Value}|}{\text{Exact Value}} \tag{11}$$

$$\mathrm{PR} = 1 - \mathrm{ER} \tag{12}$$

The proportion of the mean ED for all inputs relative to the circuit's maximum input is known as the NED. Table 9 presents the error measurements of both existing and suggested AxAs and AxSCs, respectively. The newly suggested AxSCs exhibited reduced ERs compared with existing AxSC designs [14]. Although the proposed AxSC designs resulted in the same number of errors, the proposed AxAs produced errors roughly equivalent to those of the existing AxA designs. The new AxSCs and AxDs showed significant enhancements in all error metrics.

Table 9. Analysis of error metrics

AxAs/AxSCs	ED	ER	PR	MED	NED
AxA5	6	0.75	0.25	0.56	0.07
AxA6	5	0.625	0.375	0.48	0.06
AxA1	3	0.375	0.625	0.36	0.045
AxA10	3	0.375	0.625	0.36	0.045
Pro AxA1	5	0.625	0.375	0.48	0.06
Pro AxA2	4	0.5	0.5	0.42	0.05
AxSC4	1	0.125	0.875	0.0625	0.01
AxSC5	1	0.125	0.875	0.0625	0.01
AxSC6	1	0.125	0.875	0.0625	0.01
AxSC7	1	0.125	0.875	0.0625	0.01
AxSC8	1	0.125	0.875	0.0625	0.01
AxSC9	4	0.5	0.5	0.42	0.05
AxSC10	6	0.75	0.25	0.56	0.07

4.6. Image processing applications

This segment explores the application of precise, current, and proposed designs for approximate arithmetic operations to various image processing techniques, including image squaring, multiplication, subtraction, and division. The effectiveness of these techniques was evaluated using PSNR. A high PSNR value signifies superior performance of image processing methods [24]. Figure 3 illustrates the results of self-multiplying "Lena" image using the exact, existing, and proposed adder designs. The resulting images displayed varying levels of contrast scaling, where the pixel values were linearly adjusted between the upper and lower limits to modify the image contrast. Each image is accompanied by its corresponding PSNR value, as shown in the image. For image squaring, PSNR values are reported as 35.5 (AxA5), 34.1 (AxA6), 37.0 (AxA1), 36.2 (AxA10), 39.7 (Proposed AxA1), and 40.9 (Proposed AxA2). This demonstrates that the proposed AxA1 and AxA2 achieve higher values than those of existing designs.

Figure 4 shows the multiplication of the "Airplane" and "Cameraman" images, displaying the results for the precise, current, and suggested designs. The PSNR values obtained were 34.48 (AxA5), 36.22 (AxA6), 36.12 (AxA1), 35.4 (AxA10), 38.8 (Proposed AxA1), and 41.9 (Proposed AxA2), further indicating the superior performance of the proposed designs.

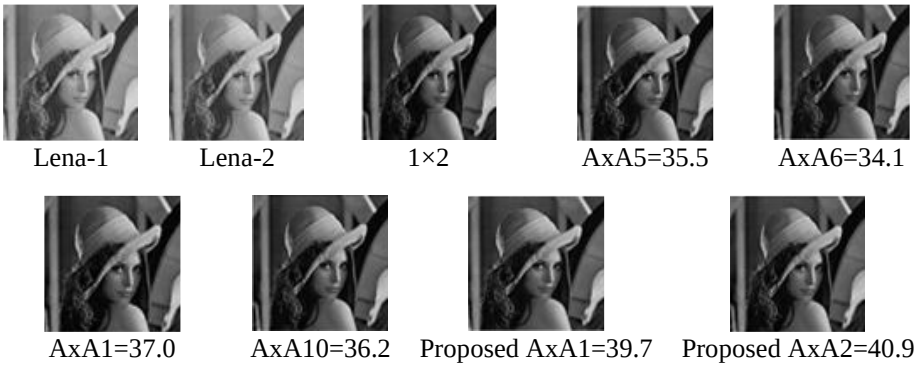


Figure 3. Image squaring

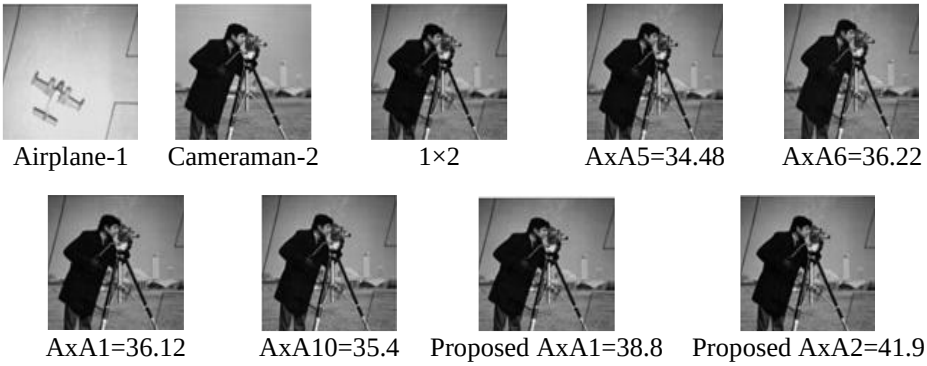


Figure 4. Image multiplication

Motion detection involves identifying the introduction or removal of objects from an environment. This procedure plays a vital role in essential operations across numerous computer vision applications, like monitoring video footage and interpreting sign language. Several techniques for detecting motion using simple image subtraction can be found in [25], [26]. Background subtraction is a common approach employed with stationary camera systems. The core idea behind these techniques is to develop a representation of a stationary scene, referred to as the background, and compare each frame in the sequence to identify the motion, which is referred to as the foreground [27].

Figure 5 shows the identification of the toy truck's movement through the subtraction of "Images 1 and 2". The resulting images, along with corresponding PSNR values, are displayed in Figure 5, providing sufficient information for the further development of motion detection applications, such as determining the object speed and detecting new object entry. The AxSCs yielded PSNR values of 24.3 (AxSC4), 24.32 (AxSC5), 24.4 (AxSC6), 24.30 (AxSC7), 25.3 (AxSC8), 24.39 (AxSC9), and 24.2 (AxSC10). The AxSC8 design, which has errors comparable to those of existing techniques, outperformed them by achieving a higher value than the existing AxSC4-7 designs.

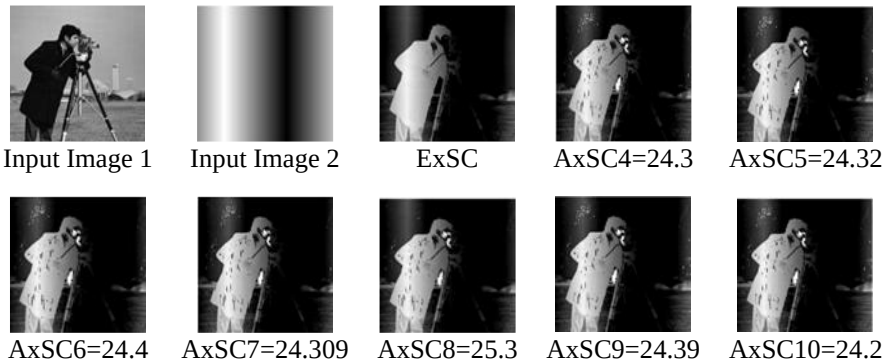


Figure 5. Image background subtraction

Finally, image division was performed using all the AxDs, and the calculated PSNR values are illustrated in Figure 6. The PSNR values for image division were 28.52 for AxD4, 25.6 for AxD5, 26.14 for AxD6, 23.37 for AxD7, 34.71 for AxD8, 24.33 for AxD9, and 24.15 for AxD10. Hence, AxD8, with errors close to those of the existing techniques, achieved a better PSNR than its counterparts.

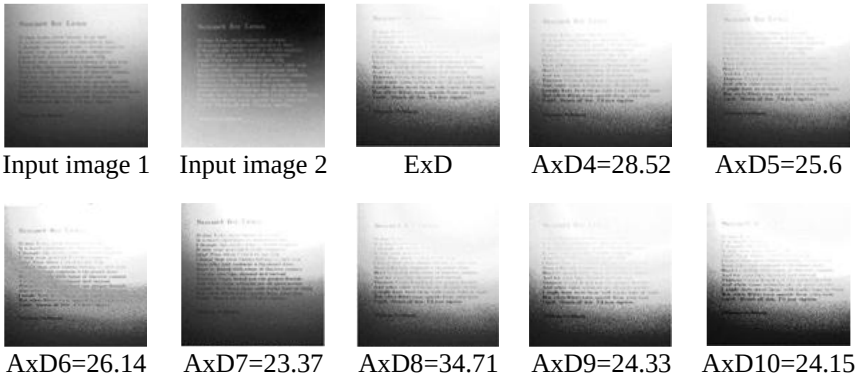


Figure 6. Image division

5. CONCLUSION

This study introduces an efficient approximate circuit design of Adders and Subtractors for effective computing based on reducing power consumption. These designs are characterized by low complexity, small area, and propagation delay. Conventional multipliers, such as the WTM and Array Dividers, are employed with the proposed AxAs and AxSCs, respectively, in the middle phases of computing, resulting in error-tolerant and energy-efficient designs. The proposed AxAs and AxSCs demonstrated superior performance by consuming less power, shorter delays, and took up less space. On average, the proposed AxAs improved by 23% and AxSCs by 38% in all the metrics mentioned. When the proposed designs were applied to image processing techniques, higher PSNR values were obtained than those of the existing approximate designs. As an emerging paradigm, approximate computation has considerable potential in energy-efficient and fault-tolerant systems.

FUNDING INFORMATION

Authors state no funding involved.

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Anusha Gorantla	✓	✓	✓	✓	✓	✓		✓	✓		✓			
Vamsidhar Anagani		✓		✓		✓	✓		✓	✓	✓			
Venkata Sridhar Gummadidala					✓	✓	✓			✓		✓		
Ramesh Babu Geddam	✓			✓	✓			✓		✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

DATA AVAILABILITY

Data availability does not apply to this paper, as no new data were created or analyzed in this study.

REFERENCES

[1] M. R. Taheri, A. Arasteh, S. Mohammadyan, A. Panahi, and K. Navi, "A novel majority based imprecise 4:2 compressor with respect to the current and future VLSI industry," *Microprocessors and Microsystems*, vol. 73, 2020, doi: 10.1016/j.micpro.2019.102962.

[2] M. M. Dominic Savio and T. Deepa, "Design of higher order multiplier with approximate compressor," *Proceedings of CONECCT 2020 - 6th IEEE International Conference on Electronics, Computing and Communication Technologies*, 2020, doi: 10.1109/CONECCT50063.2020.9198611.

[3] G. Anusha and P. Deepa, "Design of approximate adders and multipliers for error tolerant image processing," *Microprocessors and Microsystems*, vol. 72, 2020, doi: 10.1016/j.micpro.2019.102940.

[4] R. Marimuthu, Y. Elsie Rezinold, and P. S. Mallick, "Design and analysis of multiplier using approximate 15-4 compressor," *IEEE Access*, vol. 5, pp. 1027–1036, 2017, doi: 10.1109/ACCESS.2016.2636128.

[5] M. H. Moaiyeri, F. Sabetzadeh, and S. Angizi, "An efficient majority-based compressor for approximate computing in the nano era," *Microsystem Technologies*, vol. 24, no. 3, pp. 1589–1601, 2018, doi: 10.1007/s00542-017-3587-2.

[6] S. Venkatachalam and S. B. 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, doi: 10.1109/TVLSI.2016.2643639.

[7] R. Marimuthu, D. Bansal, S. Balamurugan, and P. S. Mallick, "Design of 8-4 and 9-4 compressors forhigh speed multiplication," *American Journal of Applied Sciences*, vol. 10, no. 8, pp. 893–900, 2013, doi: 10.3844/ajassp.2013.893.900.

[8] C. Li, W. Luo, S. S. Sapatnekar, and J. Hu, "Joint precision optimization and high level synthesis for approximate computing," *Proceedings - Design Automation Conference*, vol. 2015-July, 2015, doi: 10.1145/2744769.2744863.

[9] R. Jothin, M. P. Mohamed, and C. Vasanthanayaki, "High performance compact energy efficient error tolerant adders and multipliers for 16-bit image processing applications," *Microprocessors and Microsystems*, vol. 78, 2020, doi: 10.1016/j.micpro.2020.103237.

[10] K. Manikantta Reddy, M. H. Vasantha, Y. B. Nithin Kumar, and D. Dwivedi, "Design and analysis of multiplier using approximate 4-2 compressor," *AEU - International Journal of Electronics and Communications*, vol. 107, pp. 89–97, 2019, doi: 10.1016/j.aeue.2019.05.021.

[11] D. Esposito, G. Di Meo, D. De Caro, A. G. M. Strollo, and E. Napoli, "Design of low-power approximate LMS filters with precision-scalability," *Lecture Notes in Electrical Engineering*, vol. 550, no. 9783030119720, pp. 237–243, 2019, doi: 10.1007/978-3-030-11973-7_27.

[12] G. Di Meo, D. De Caro, E. Napoli, N. Petra, and A. G. M. Strollo, "Low-power implementation of LMS adaptive filters using scalable rounding," *ICECS 2020 - 27th IEEE International Conference on Electronics, Circuits and Systems, Proceedings*, 2020, doi: 10.1109/ICECS49266.2020.9294848.

[13] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, 2013, doi: 10.1109/TCAD.2012.2217962.

[14] L. Chen, J. Han, W. Liu, and F. Lombardi, "Design of approximate unsigned integer non-restoring divider for inexact computing," *Proceedings of the ACM Great Lakes Symposium on VLSI, GLSVLSI*, vol. 20-22-May-2015, pp. 51–56, 2015, doi: 10.1145/2742060.2742063.

[15] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4169–4182, 2018, doi: 10.1109/TCSI.2018.2839266.

[16] 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 Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 9, pp. 3021–3034, 2020, doi: 10.1109/TCSI.2020.2988353.

[17] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2015, doi: 10.1109/TC.2014.2308214.

[18] E. Jagadeeswara Rao, K. Jayaram Kumar, and T. V. Prasad, "Design of high speed wallace tree multiplier using 8-2 and 4-2 adder compressors," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 4, pp. 2386–2390, 2018, doi: 10.14419/ijet.v7i4.12261.

[19] P. N. V. K. Hasini and T. K. Murthy, "A Novel high-speed transistorized 8x8 Multiplier using 4-2 Compressors," *International Journal of Engineering Research and General Science*, vol. 3, no. 2, pp. 359–365, 2015, [Online]. Available: www.ijergs.org.

[20] M. Lu, *Arithmetic and logic in computer systems*. Wiley, 2004.

[21] R. Mishra, "An efficient VLSI architecture for a serial divider," *Proceedings of 2nd International Conference on 2017 Devices for Integrated Circuit, DevIC 2017*, pp. 482–486, 2017, doi: 10.1109/DEVIC.2017.8073996.

[22] A. Gorantla and P. Deepa, "Design of approximate subtractors and dividers for error-tolerant image processing applications," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 35, no. 6, pp. 901–907, 2019, doi: 10.1007/s10836-019-05837-5.

[23] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, 2013, doi: 10.1109/TC.2012.146.

[24] T. U. Swathi Krishna, K. S. Riyas, Y. Premson, and R. Sakthivel, "15-4 approximate compressor-based multiplier for image processing," *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, ICOEI 2018*, pp. 671–675, 2018, doi: 10.1109/ICOEI.2018.8553734.

[25] A. Manzanera and J. C. Richefeu, "A new motion detection algorithm based on Σ-Δ background estimation," *Pattern Recognition Letters*, vol. 28, no. 3, pp. 320–328, 2007, doi: 10.1016/j.patrec.2006.04.007.

[26] S. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Visual Communications and Image Processing 2004*, vol. 5308, p. 881, 2004, doi: 10.1117/12.526886.

[27] H. A. F. Almurib, T. N. Kumar, and F. Lombardi, "Approximate DCT image compression using inexact computing," *IEEE Transactions on Computers*, vol. 67, no. 2, pp. 149–159, 2018, doi: 10.1109/TC.2017.2731770.

BIOGRAPHIES OF AUTHORS



Anusha Gorantla received her Bachelor of Technology in Electronics and Communication Engineering (ECE) from JNTU, Hyderabad in 2006. She received Master's VLSI Design in 2008 and a Ph.D. in Information and Communication Engineering in 2018 from Anna University, Chennai, India. Now, she works as an Associate Professor, at the Department of Electronics and Communication Engineering, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India. Her research includes low-power VLSI design, approximate computing, and image processing. She can be contacted at email: anushagorantla3@gmail.com.



Vamsidhar Anagani received his bachelor of Technology in Electronics and Communication Engineering (ECE) from Jawaharlal Nehru Technological University, Kakinada, India in 2000 and Master's in Digital Systems and Communications from National Institute of Technology, Calicut, India in 2003. He received his Ph.D. in Electronics and Communication Engineering from Andhra University, Visakhapatnam, India in 2018. He is currently a Professor of Electronics and Communication Engineering at Raghu Engineering College, Visakhapatnam, India. He has more than 22 years of teaching and research experience in various academic institutions. His areas of research include wireless communications, signal, image, and speech processing, VLSI, and machine learning. He can be contacted at email: vamsianagani@gmail.com.



Venkata Sridhar Gummadidala received his Ph.D, M.E and B.E. from Andhra University, Visakhapatnam, India in 2017, 2005, and 1998 respectively. He is currently working as a Professor in the Department of Electronics and Communication Engineering, Dadi Institute of Engineering and Technology (A), Anakapalle, India. He has published 20 articles in various reputed national/international journals and conferences. His research interests include biomedical signal processing, prosthetic devices, and medical electronics. He was a reviewer at the 10th International Conference on Electronics, Communications, and Networks (Oct. 25-28, 2020, Seoul, Republic of Korea), ICMSEA 2018, and ICMSEA 2017. He is a senior member of IEEE and a life member of IETE. He can be contacted at email: sridhar1209@gmail.com.



Ramesh Babu Geddam received his B.Tech in Electronics and Communications Engineering from Nagarjuna University, Andhra Pradesh (1994-1998), and obtained his M.E from PSG College of Technology, Coimbatore, Tamil Nadu (2001) and Ph.D in Electronics and Communication Engineering from Osmania University, Hyderabad, India (2014). He is currently a Professor of Electronics and Communication Engineering at Raghu Engineering College, Visakhapatnam, India. He has over 18 Years of teaching and industrial experience. He has published more than 23 publications, which include journal papers and conference papers. His research interests are in the areas of signal processing, non-stationary signal processing, speech processing, and image processing. He is a Life Member of Institution of Engineers (IE) and ISTE. He can be contacted at email: drgramesh24@gmail.com.