International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 14, No. 3, December 2025, pp. 1189~1198
ISSN: 2252-8776, DOI: 10.11591/ijict.v14i3.pp1189-1198

Efficient design of approximate carry-based sum calculating full adders for error-tolerant applications

Badiganchela Shiva Kumar[1], Galiveeti Umamaheswara Reddy[2]
[1]SVU College of Engineering, Sri Venkateswara University, Tirupati, India
[2]Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, India

Article Info

Article history:

Received Nov 1, 2024
Revised Jun 19, 2025
Accepted Jul 1, 2025

Keywords:

Approximate computing adder
Approximate sum
Error tolerant
Full adder
Hardware complexity
Verilog HDL

ABSTRACT

Approximate computing is an innovative circuit design approach which can be applied in error-tolerant applications. This strategy introduces errors in computation to reduce an area and delay. The major power-consuming elements of full adder are XOR, AND, and OR operations. The sum computation in a conventional full adder is modified to produce an approximate sum which is calculated based on carry term. The major advantage of a proposed adder is the approximation error does not propagate to the next stages due to the error only in the sum term. The proposed adder was coded in verilog HDL and verified for different bit sizes. Results show that the proposed adder reduces hardware complexity with delay requirements.

Corresponding Author:

Badiganchela Shiva Kumar
SVU College of Engineering, Sri Venkateswara University
Tirupati, Andhra Pradesh, India
Email: bshivakumaries@gmail.com

1. INTRODUCTION

Approximate computing is a recent emerging technology that has a promising methodology for an energy-efficient model in digital systems [1]. Approximate computing is the collaboration of multiple systems and applications that tolerate some error occurring in the output or the quality of the system in the obtained result. These methods have omitted the requirement of a fully precise or completely exact process of circuits. These approximate computing techniques are support significantly for energy efficiency improvement respectively.

The approximation technique is applied at the level of architecture or the circuital blocks [2]-[3]. These techniques are categorized according to the hardware components that are affected. Some of the main blocks of architecture are functional units, memory, circuit design, relaxed fault tolerance, microarchitecture and stochastic computing.

The approximate computing is applied in the many electronic digital circuits which are carried from the basic element of logic gates of adders and multipliers. This is classified into several categories namely approximate full adders, multiple-bit approximate adders, approximate multipliers and approximate logic synthesis respectively.

Adders are frequently used elements in arithmetic systems [4]. There are two ways to design low-power adders: adders with gate-level modifications and adders with circuit-level modifications. The adders with circuit level alternation include different circuit styles for their construction. The popular low power adder circuit style methods are adiabatic, transmission gate and pass transistor logics. In gate-level modifications, the number of gates required for the adder construction is simplified using K-Maps.

There are many quality metrics defined for measuring the accuracy of Approximate computation [5], [6]. Many of the widely employed measures are: minimum acceptable accuracy (MAA), amplitude accuracy (ACC amp), information accuracy (ACC inf), normalized error distance (NED), mean error distance (MED), error significance (ES), error rate (ER), and overall error magnitude. Traditionally, all such parameters are predictions which have been done based on the Monte-Carlo simulations. In addition, most of them have long-term averages or significant values. The precision of approximation computation can be better defined by the probability mass function (PMF) of the error value (absolute or signed, since required), as it displays all reasonable error values and their respective orders of magnitude. Each of the aforementioned parameters can be measured from the PMF. Existing approximate computing methods often overlook the optimization of error propagation in full adders for large-scale applications. There is a need for further investigation into optimizing error propagation in approximate full adders, particularly in large-scale applications.

In this paper, a new type of approximate adder is proposed and analysed for different bit sizes. For fair analysis, the proposed adder is applied in image processing applications of discrete cosine transform (DCT). This work is structured as follows. In section 2 explains the existing works. The design of the proposed adder and its transform implementations are given in section 3. In section 4 illustrates the experimental results. Finally, the conclusion is described in section 5.

2. RELATED WORK

Amirafshar *et al.* [7] constructed a new ripple carry adder with the logic of carry disregarding. The new logic level modification is proposed to reduce an overall propagation delay. The proposed adder reduces area power and delay by about 8.56%, 19.5%, and 12.4% when compared to other adders.

In their study, Fan *et al.* [8] present an approximate adder based on transistor-level modifications. The transistor acts as a switch to control the carry propagation. Experimental results on synopsis compiler results show that the proposed adder achieves 6.3% area reduction as an average of varying bit sizes.

The multi-bit adder using a parallel prefix adder is proposed by Stefanidis *et al.* [9]. The concept of carry killing is applied to mitigate a critical path delay. The proposed adder is implemented in a finite impulse response filter and verified for signal processing applications. Results show the proposed adder shows higher efficiency with acceptable noise limits in filter application.

Kavand *et al.* [10], a new compressor structure is proposed by the author for implementing an approximate multiplier. The proposed compressor is applied in vector merging adder stages in the multiplier for final addition calculation. The number of slice and logic counts is very low in comparison with other compressor structures.

The hybrid adder for low power application is introduced by Seo and Kim [11]. The hybrid adder includes both accurate and inaccurate adders to compute final addition results. The accurate adder produces exact results without any truncations. In the inaccurate adder part, the adders with a minimum number of gates are used for sum calculation.

Jha *et al.* [12] proposed an accuracy configurable approximate adder for error-tolerant applications. The accuracy level of the adder circuit can be configured based on the terms that are used for sum calculation. In addition, the correction logic is introduced to reduce an approximation error.

Zhang *et al.* [13] proposed a threefold approximate multiplier for lossy applications. It involves compressors, correction units and fixed approximations. The correction unit varies the correction bits based on the compressor error levels. Results on image processing purposes show that the proposed multiplier achieves minimum area delay product with higher signal-to-noise ratio (SNR) values.

Chen *et al.* [14] proposed a modified booth encoder for approximate computing. The encoding bits in the booth table are approximated to perform selective truncation. The proposed multiplier is applied in the machine learning algorithm for efficiency evaluation.

Da Rosa *et al.* [15] proposed a parallel adder using the concept of carry propagation and carry generation. The proposed adder is used to design approximate convolutional neural network architecture for pattern recognition. By using the proposed adder, the number of adder cost requirements is reduced considerably. In their study, Chu *et al.* [16] presented an approximate adder using quantum-dot cellular automata technology. The majority of logic minimization is utilized to design an approximate circuit. The proposed adder was analysed in terms of the number of dots, power and delay requirements.

Santiago *et al.* [17] proposed a carry-aware approximate adder for multi-bit addition. The elimination of the least significant bit is carried out based on the number of ones present in the input operands. In image compression applications, the proposed adder shows a higher accuracy level than other adders. Zhang *et al.* [18], the author proposed a new approach for partial product generation in booth multiplier. It applies machine learning algorithms to produce a partial product. The proposed multiplier is

tested in feed-forward neural network implementations. The dual error correction mechanism is proposed by Aizaz and Khare [19] for-booth multiplier design. The error correction mechanism uses k-map simplification to reduce the number of gate requirements. Implementation results on image processing application show the proposed multiplier achieves 94.36% accuracy with an error rate of 0.02%.

The concept of machine learning is applied in approximate computing by Liu *et al.* [20]. The approximating gates or blocks in a circuit are identified with learning models to get a minimum approximation error when removed. The additional effort or computational resources are utilized for learning models. In their study, Roy *et al.* [21] proposed a hybrid adder which includes the approximate least significant bit part and the accurate most significant bit part. In the approximate part, the truncation with a correction mechanism is introduced to reduce the overall area delay product. The carry chain division mechanism is used by Ebrahimi-Azandaryani *et al.* [22] for low-power adder design. Based on operand bits, the carry chain is divided into different lengths to reduce overall critical delay. Likewise, the new strategy is proposed by Frustaci *et al.* [23] for approximate adder and multiplier design. Similar way, the Wu *et al.* [24] derived new metrics for assessing the performance of approximate circuits.

Wu *et al.* [24], the authors introduce a reverse carry concept for multi-bit adder design. The carry is generated from the most significant bit to the least significant bit to reduce an approximation error. The proposed adder is implemented in DCT architecture for domain conversion applications.

Alan and Henkel [25] proposed an approximate carry look adder by modifying the carry prediction term. By integrating a multiplexer, the adder can be configured to either approximate or accurate modes. Schlachter *et al.* [26] introduce a pruning technique for approximate circuit design. In the pruning process, the logical gates are selectively removed based on the error level. The final gate-level netlist is created after the pruning process.

Tsai *et al.* [27] proposed a memory-based approximation technique for adder and multiplier design. The addition and multiplication terms are stored in the look-up table to reduce computation costs. Compared to truncation schemes, the memory-based approach shows minimum approximation errors. Likewise, different adders are proposed for different image-processing applications [28].

3. PROPOSED SYSTEM

In this work, a new approximate adder is proposed based on gate-level modifications. The major power-hungry source of adder elements is XOR gate operations. This work modifies sum term calculation based on carry term by eliminating the XOR operation. The Boolean function for the conventional full adder is stated as follows:

$$\text{Sum} = a \text{ XOR } b \text{ XOR } c; \quad (1)$$

$$\text{Carry} = (a \text{ AND } b) \text{ OR } (b \text{ AND } c) \text{ OR } (c \text{ AND } a); \quad (2)$$

The expression for the proposed adder is as follows:

$$\text{Carry} = (a \text{ AND } b) \text{ OR } (b \text{ AND } c) \text{ OR } (c \text{ AND } a); \quad (3)$$

$$\text{Sum} = \text{NOT } (\text{Carry AND } b); \quad (4)$$

The sum term is derived from the carry. By negating the logical AND of the carry-out and input bit A, the sum is calculated in a straightforward manner without the need for XOR gates. This method introduces a controlled approximation into the sum calculation. The structure of conventional and proposed full adder is shown in Figure 1, Figures 1(a) and 1(b). The functionality of the original and proposed full adder is given in Table 1.

Table 1. Truth table of proposed adder

A	B	C_{in}	Sum	Carry	$Carry_{app}$
0	0	0	0	0	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	0	1	0
1	1	0	0	1	0
1	1	1	1	1	0

A key advantage of this design is that the error is confined only to the sum term, and it does not propagate through the carry chain. Specifically, out of the 8 possible input combinations for a 3-bit adder, only 3 combinations result in an erroneous sum. This controlled error rate is acceptable in many approximate computing applications where slight inaccuracies are tolerable.

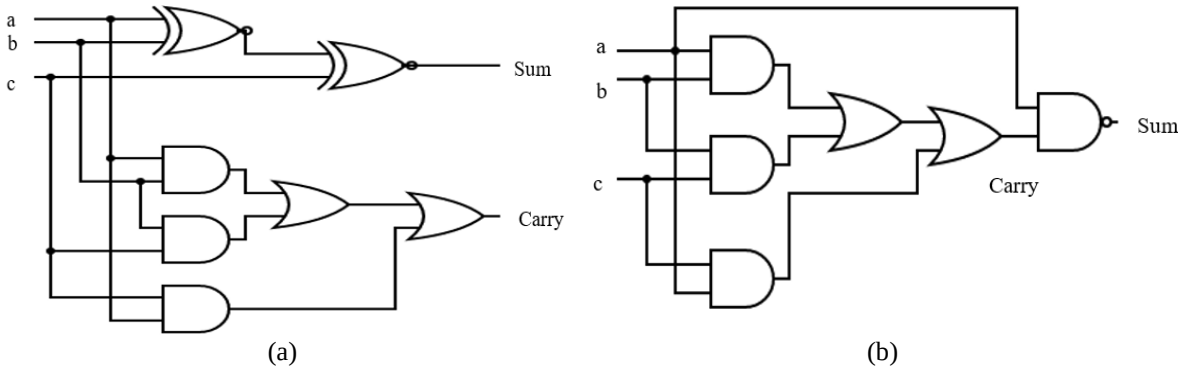

Figure 1. Structure of conventional and proposed full (a) conventional adder and (b) proposed adder

Approximate adder-based discrete cosine transforms, in image processing, the DCT is widely used for image compression applications such as JPEG. The DCT transforms spatial domain data into frequency domain data to detect a significant coefficient that can be compressed effectively. The approximate adders are used in DCT to reduce power consumption, area, and delay. The DCT formula for an 8×8 block is:

$$DCT(u,v) = \frac{1}{4}c(u)c(v)\sum_{x=0}^{7}\sum_{y=0}^{7} I(x,y)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right] \quad (5)$$

Where c(u) and c(v) are the normalization factors. Initially, the images are converted into gray-scale images. Then, the entire image is divided into 8×8 blocks. The proposed approximate adder is applied in additional stages of DCT computation. The results were verified in terms of SNR rates.

4. RESULT AND DISCUSSION

The proposed adder is coded in verilog HDL and executed using the Xilinx software environment. The proposed adder with varying bit sizes is implemented in the Spartan3E FPGA family with the device XC3S100E. The corresponding RTL schematic and gate-level netlist are shown in Figure 2. The visualization of functional verification is given in Figure 3. For a fair comparison, the proposed adder is compared with previously proposed adders.

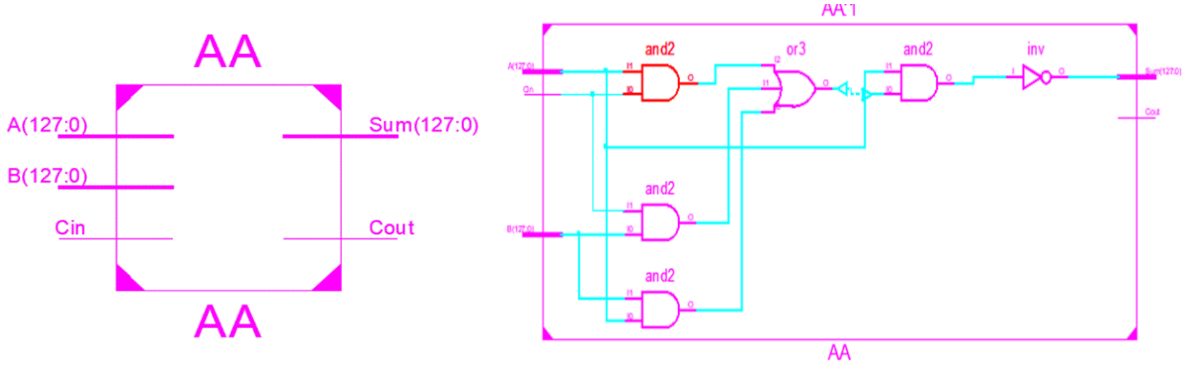

Figure 2. RTL and gate level netlist of adder

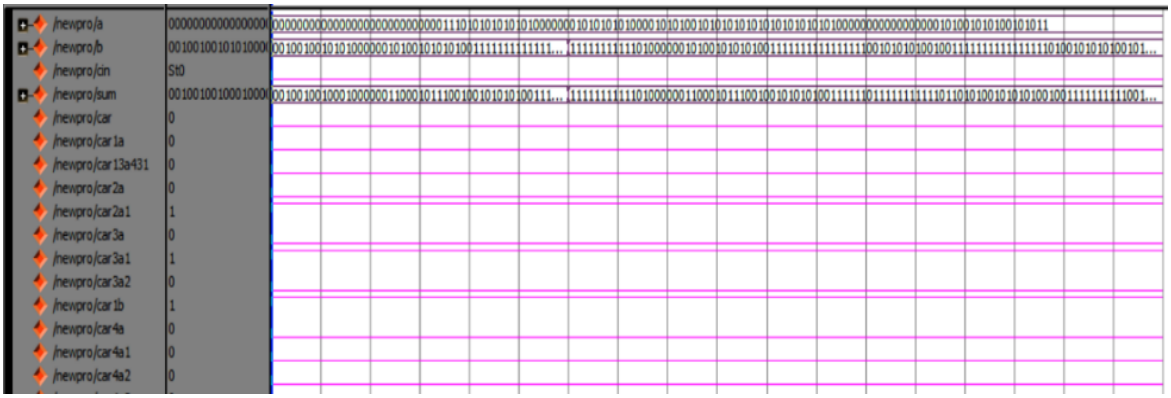

Figure 3. Simulated waveform

The obtained slice and LUT values for varying bit sizes are given in Table 2. The slice and LUT values increase linearly with increasing bit sizes. The proposed adder shows a balanced and optimized performance in terms of all parameters when compared to other adders. The mirror approximate adder proposed by Gupta *et al.* [29] shows the highest delay and area consumption. The approximate adder based on the majority logic proposed by W. Liu shows moderate performance in terms of area and delay.

The RTL view and gate-level netlist of the proposed DCT implementation is shown in Figure 4. The visualization of functional verification is given in Figure 5. The values obtained for implementation are given in Table 3.

Table 2. Performance analysis

	Bit size	SLICE	LUT	DELAY
Gupta *et al.* [29]	16	24	46	31.9
	32	58	96	70.56
	64	124	218	103.6
	128	248	420	212.48
Venkatachalam and Ko [30]	16	24	46	34.56
	32	53	90	58.478
	64	118	200	90.67
	128	244	390	154.89
Strollo *et al.* [31]	16	19	42	35.89
	32	49	80	42.3
	64	112	192	85.6
	128	240	383	132.678
Liu *et al.* [32]	16	19	38	22.13
	32	46	82	35.43
	64	102	171	68.12
	128	211	342	118.8
Proposed adder	16	19	34	15.457
	32	45	80	27.836
	64	96	167	52.162
	128	193	336	98.8

Table 3. FIR implementation results

Parameter	DCT with majority-based approximate adder	DCT with proposed adder
slice	687	453
LUT	1012	856
Delay	55.6	42.89

The DCT with the majority adder uses 687 slices. In contrast, the DCT with the proposed adder shows improved efficiency by using only 453 slices. The proposed adder achieves a 12% reduction in slice usage. The DCT filter with the majority adder requires 1012 LUTs. On the other hand, the DCT with the proposed adder consumes 856 LUTs. The proposed adder achieves a 15% reduction in LUT usage. The majority of adder-based DCT shows a delay of 55.6 units. However, the proposed adder-based DCT notably outperforms this with a delay of only 42.89 units.

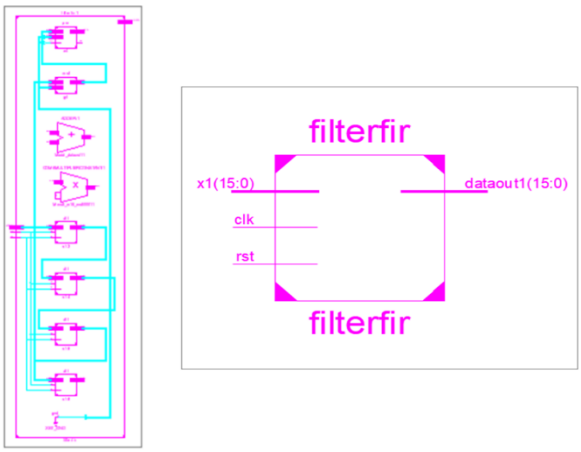

Figure 4. RTL and gate level netlist of DCT

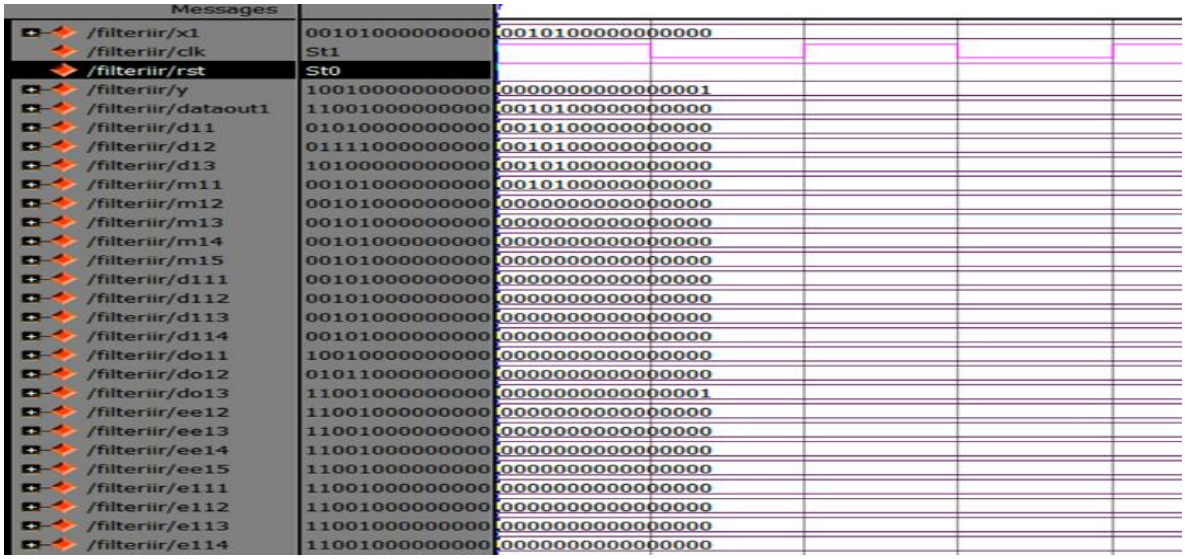

Figure 5. Simulated waveform

A peak signal-to-noise ratio (PSNR) is used as a metric for the performance assessment to compare the original image against the approximate one. The PSNR of the image is represented in dB can be computed as,

$$PSNR=10\log_{10}((MN*255^2)/(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[P(i,j)-P'(i,j)]^2) \quad (6)$$

The existing and the proposed adders are implemented in image processing applications such as smoothening and sharpening of the images. The implementation results are given in Table 4.

Table 4. PSNR comparison of image

Application	DCT	
Benchmark	Majority-based approximate adder	Proposed
Lena	12.2	13.3
Tiffany	6.5	8.0
House	12.5	13.5
Cameraman	13	14.4
Moon surface	10.6	10.8
Peppers	13	15
Tulips	11.5	14.1

The performance of the proposed adder is compared against a majority-based approximate adder. For the Lena image, the majority-based approximate adder achieves a PSNR of 12.2. The proposed adder scores higher than the majority-based adder about 13.3. Similarly, for the Tiffany image, the proposed adder improves the PSNR from 6.5 to 8.0. For the Peppers image, the PSNR increases significantly from 13 to 15, and for the Tulips image, the PSNR improves from 11.5 to 14.1. These results indicate that the proposed adder consistently provides better performance across various images compared to the majority-based approximate adder. The improvements suggest that the proposed design enhances error resilience while maintaining or even improving the quality of the processed images, demonstrating its effectiveness in error-tolerant applications. The input and corresponding output images are shown in Figure 6.

Figure 6. Input and corresponding DCT output images

5. CONCLUSION

Approximate computing has gained more attention in the field of multimedia and real-time applications. In this paper, we presented a novel approximate adder design that significantly reduces hardware complexity and power consumption. It eliminates the need for XOR operations in conventional approximate adders. This approach confines errors to the sum term only which assures no error propagation through the carry chain. The simplified architecture and controlled error rate make this approximate adder a

promising solution for energy-efficient and high-performance approximate computing applications. Future research will focus on optimizing the proposed approximate carry-based sum calculating full adders to further reduce energy consumption and hardware complexity. Additionally, incorporating advanced error correction techniques and exploring the use of machine learning for adaptive error tolerance in different applications could improve the overall performance and accuracy.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support and guidance provided by the Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, India, during the research work.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Badiganchela Shiva Kumar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Galiveeti Umamaheswara Reddy	✓	✓		✓		✓	✓			✓		✓	✓	

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

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

Not applicable, as the study did not involve human participants.

ETHICAL APPROVAL

Not applicable, as the study did not involve human or animal subjects.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] S. Mittal, "A survey of techniques for approximate computing," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–33, May 2016, doi: 10.1145/2893356.

[2] Q. Zhang, F. Yuan, R. Ye, and Q. Xu, "ApproxIt: an approximate computing framework for iterative methods," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, Jun. 2014, pp. 1–6, doi: 10.1109/dac.2014.6881424.

[3] B. Sakthivel and A. Padma, "Area and delay efficient GDI based accuracy configurable adder design," *Microprocessors and Microsystems*, vol. 73, p. 102958, Mar. 2020, doi: 10.1016/j.micpro.2019.102958.

[4] M. Ringenburg, A. Sampson, I. Ackerman, L. Ceze, and D. Grossman, "Monitoring and debugging the quality of results in approximate programs," *ACM SIGPLAN Notices*, vol. 50, no. 4, pp. 399–411, May 2015, doi: 10.1145/2775054.2694365.

[5] M. Usharani, B. Sakthivel, K. Jayaram, and R. Renugadevi, "Design of logically obfuscated memory and arithmetic logic unit for improved hardware security," *Intelligent Automation and Soft Computing*, vol. 33, no. 3, pp. 1665–1675, 2022, doi: 10.32604/iasc.2022.023284.

[6] M. M. A. daRosa, L. Antonietti, R. Lopes, E. Barros, E. A. C. daCosta, and R. Soares, "FALSAx: an integrated framework for accuracy and logic synthesis estimation of approximate adders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 72, no. 4, pp. 1679–1692, Apr. 2025, doi: 10.1109/TCSI.2024.3511383.
[7] N. Amirafshar, A. S. Baroughi, H. S. Shahhoseini, and N. Taherinejad, "Carry disregard approximate multipliers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 12, pp. 4840–4853, Dec. 2023, doi: 10.1109/TCSI.2023.3306071.
[8] X. Fan, T. Zhang, H. Li, H. Liu, S. Lu, and J. Han, "DACA: dynamic accuracy-configurable adders for energy-efficient multi-precision computing," *IEEE Transactions on Nanotechnology*, vol. 22, pp. 400–408, 2023, doi: 10.1109/TNANO.2023.3297325.
[9] A. Stefanidis, I. Zoumpoulidou, D. Filippas, G. Dimitrakopoulos, and G. C. Sirakoulis, "Synthesis of approximate parallel-prefix adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 11, pp. 1686–1699, Nov. 2023, doi: 10.1109/TVLSI.2023.3287631.
[10] N. Kavand, A. Darjani, S. Rai, and A. Kumar, "Design of energy-efficient RFET-based exact and approximate 4:2 compressors and multipliers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 9, pp. 3644–3648, Sep. 2023, doi: 10.1109/TCSII.2023.3275983.
[11] H. Seo and Y. Kim, "A low latency approximate adder design based on dual sub-adders with error recovery," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 3, pp. 811–816, Jul. 2023, doi: 10.1109/TETC.2023.3270963.
[12] C. K. Jha, A. Nandi, and J. Mekie, "Single exact single approximate adders and single exact dual approximate adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 7, pp. 907–916, Jul. 2023, doi: 10.1109/TVLSI.2023.3268275.
[13] M. Zhang, S. Nishizawa, and S. Kimura, "Area efficient approximate 4-2 compressor and probability-based error adjustment for approximate multiplier," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 5, pp. 1714–1718, May 2023, doi: 10.1109/TCSII.2023.3257852.
[14] K. Chen, C. Xu, H. Waris, W. Liu, P. Montuschi, and F. Lombardi, "Exact and approximate squarers for error-tolerant applications," *IEEE Transactions on Computers*, vol. 72, no. 7, pp. 2120–2126, 2023, doi: 10.1109/TC.2022.3228592.
[15] M. M. E. A. Da Rosa, G. Paim, P. U. L. Da Costa, E. A. C. Da Costa, R. I. Soares, and S. Bampi, "AxPPA: approximate parallel prefix adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 1, pp. 17–28, Jan. 2023, doi: 10.1109/TVLSI.2022.3218021.
[16] Z. Chu, C. Shang, T. Zhang, Y. Xia, L. Wang, and W. Liu, "Efficient design of majority-logic-based approximate arithmetic circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 12, pp. 1827–1839, Dec. 2022, doi: 10.1109/TVLSI.2022.3210252.
[17] F. J. H. Santiago, H. Jiang, H. Amrouch, A. Gerstlauer, L. Liu, and J. Han, "Characterizing approximate adders and multipliers for mitigating aging and temperature degradations," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 11, pp. 4558–4571, Nov. 2022, doi: 10.1109/TCSI.2022.3193928.
[18] T. Zhang *et al.*, "Design of majority logic-based approximate booth multipliers for error-tolerant applications," *IEEE Transactions on Nanotechnology*, vol. 21, pp. 81–89, 2022, doi: 10.1109/TNANO.2022.3145362.
[19] Z. Aizaz and K. Khare, "Area and power efficient truncated booth multipliers using approximate carry-based error compensation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 2, pp. 579–583, Feb. 2022, doi: 10.1109/TCSII.2021.3094910.
[20] W. Liu, T. Zhang, E. McLarnon, M. Orneill, P. Montuschi, and F. Lombardi, "Design and analysis of majority logic-based approximate adders and multipliers," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1609–1624, Jul. 2021, doi: 10.1109/TETC.2019.2929100.
[21] A. S. Roy, R. Biswas, and A. S. Dhar, "On fast and exact computation of error metrics in approximate LSB adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 876–889, Apr. 2020, doi: 10.1109/TVLSI.2020.2967149.
[22] F. Ebrahimi-Azandaryani, O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Block-Based carry speculative approximate adder for energy-efficient applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 137–141, Jan. 2020, doi: 10.1109/TCSII.2019.2901060.
[23] F. Frustaci, S. Perri, P. Corsonello, and M. Alioto, "Energy-quality scalable adders based on nonzeroing bit truncation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 964–968, Apr. 2019, doi: 10.1109/TVLSI.2018.2881326.
[24] Y. Wu, Y. Li, X. Ge, Y. Gao, and W. Qian, "An efficient method for calculating the error statistics of block-based approximate adders," *IEEE Transactions on Computers*, vol. 68, no. 1, pp. 21–38, Jan. 2019, doi: 10.1109/TC.2018.2859960.
[25] T. Alan and J. Henkel, "Probability-driven evaluation of lower-part approximation adders," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 1, pp. 204–208, Jan. 2022, doi: 10.1109/TCSII.2021.3093984.
[26] J. Schlachter, V. Camus, K. V. Palem, and C. Enz, "Design and applications of approximate circuits by gate-level pruning," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1694–1702, May 2017, doi: 10.1109/TVLSI.2017.2657799.
[27] K. L. Tsai, Y. J. Chang, C. H. Wang, and C. T. Chiang, "Accuracy-Configurable Radix-4 adder with a dynamic output modification scheme," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 8, pp. 3328–3336, Aug. 2021, doi: 10.1109/TCSI.2021.3085572.
[28] A. Amirany, G. Epperson, A. Patooghy, and R. Rajaei, "Accuracy-adaptive spintronic adder for image processing applications," *IEEE Transactions on Magnetics*, vol. 57, no. 6, pp. 1–10, Jun. 2021, doi: 10.1109/TMAG.2021.3069161.
[29] 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, Jan. 2013, doi: 10.1109/TCAD.2012.2217962.
[30] 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, May 2017, doi: 10.1109/TVLSI.2016.2643639.
[31] 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, Sep. 2020, doi: 10.1109/TCSI.2020.2988353.
[32] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 booth multipliers for error-tolerant computing," *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, Aug. 2017, doi: 10.1109/TC.2017.2672976.

BIOGRAPHIES OF AUTHORS

Badiganchela Shiva Kumar received the B.Tech. Degree in Electronics and Communication Engineering from V.I.T.S. Proddatur, Jawaharlal Nehru Technological University, Anantapur, India, in 2009, M.Tech. Degree in Digital Electronics and Communication Systems from J.N.T.U.A.C.E. Pulivendula, Jawaharlal Nehru Technological University, Anantapur, India, in 2018. He is currently pursing Research Scholar at SVU College of Engineering, Sri Venkateswara University, Tirupati, India. His current research interests include through VLSI circuit design, device modeling and VLSI system design, and VLSI signal processing. He can be contacted by email: bshivakumaries@gmail.com.

Galiveeti Umamaheswara Reddy received B.Tech. Degree in Electronics and Communication Engineering, M.Tech. in Instrumentation and Control Systems, and obtained Ph.D. from Sri Venkateswara University, Tirupati. He is a member of the ISTE, IEI, and BMESI. Currently, he is Professor at Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, Andhra Pradesh, India. He has a teaching experience of more than 28 years and has 69 technical publications in national/international journals and Conferences. His areas of interest include VLSI design and signal processing. He can be contacted by email: umaskit@gmail.com.