Efficient Image Blending Techniques Using FPGA-Based Approximate Multipliers

Sreelakshmi Vadlamudi [1,2], Syamala Yarlagadda [2] and Madhu R [3]

[1] *Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh, India.*
sreelakshmi.vadlamudi@gmail.com

[2] *Seshadri Rao Gudlavalleru Engineering College, Andhra Pradesh, India. coolsyamu@gmail.com*

[3] *Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh, India.*
madhu_ramarkula@jntucek.ac.in

Abstract.

In error-resilient computing, particularly in applications like multimedia processing, data analysis, and machine learning, absolute computational precision is often not a strict requirement. This paper introduces a novel FPGA-based design for multi-level approximate multipliers that optimally balance accuracy with key performance metrics such as power consumption, area efficiency, and processing speed. By leveraging techniques like partial product truncation and logic compression, the proposed multipliers achieve significant reductions in circuit complexity while enhancing overall efficiency. The results demonstrate that these designs effectively cater to applications where minor inaccuracies are acceptable, delivering notable improvements in power-delay-area performance compared to conventional exact multipliers.

Keywords Approximate Computing, Approximate Multipliers, Power-Efficient Computing, Circuit Complexity Reduction, Partial Product Truncation, FPGA Design.

1. INTRODUCTION

In the landscape of modern computing, many applications, particularly those in fields such as multimedia processing, data mining, and machine learning, do not require absolute precision in their calculations[1]-[4]. Instead, these applications can function effectively with results that exhibit a certain level of acceptable error [5]. This characteristic opens the door to innovative design strategies that prioritize performance metrics—such as power consumption, area efficiency, and processing speed—over strict accuracy. By embracing this paradigm, engineers can create systems that are not only faster and more efficient but also more suited to the demands of real-world applications where resource constraints are prevalent.

One of the prominent techniques for reducing power consumption in digital circuits is voltage over-scaling [6] –[8]. While this method can lead to significant energy savings, it also introduces the risk of timing-related failures, which can result in substantial and unpredictable errors, particularly affecting the most significant bits of computation. To address these challenges, the concept of approximate computing has gained traction. This approach allows for the intentional introduction of errors in the outputs of complex circuits, enabling designers to simplify logic expressions and reduce the overall resource requirements. By accepting a controlled level of inaccuracy, approximate computing can lead to considerable improvements in power efficiency and performance.

The focus of much recent research has been on the application of approximation techniques to arithmetic units, especially multipliers and adders, which are fundamental components in a wide array of digital systems[9] –[12]. Given the inherent complexity of multipliers, optimizing their design through approximation can yield significant benefits. Notably, the partial product accumulation stage

of a multiplier is where the majority of computational resources and time are consumed [13]. Therefore, enhancing this stage through innovative design techniques can lead to substantial gains in overall circuit performance.

In this paper, we propose a novel architecture for FPGA-based multi-level approximate multipliers that utilize advanced approximate logic compressors. These compressors are designed to reduce the complexity of the partial product accumulation process while maintaining an acceptable level of accuracy for applications that can tolerate some error. Our findings demonstrate that these multipliers not only achieve significant improvements in power-delay-area products but also outperform traditional exact multipliers in various performance metrics. By exploring the trade-offs between accuracy and efficiency, this work contributes to the growing body of knowledge in approximate computing, providing valuable insights for the design of high-performance, low-power digital systems.

1) It is suggested that approximate multipliers be effectively implemented on FPGAs with minimal accuracy losses by utilizing the small approximate 4-2 compressors.
2) Various proposed multiplier designs can be used to obtain design-time accuracy configurations.

3) Quality measures such as mean error distance (MED), mean relative error distance (MRED), and normalized mean error distance (NMED) are used to extensively analyze the quality of the proposed approximate multipliers.
4) Extensive simulations are carried out to get precise QMs for 8X8 multipliers.
5) Formulas for calculating the crucial metrics (MED and NMED) for multipliers with large operand sizes that are scaled up from those with lower operand sizes are derived.
6)Additionally, the trade-offs between area and PDPs are shown.
7) The benefits and efficiency of our suggested multipliers are demonstrated using image processing applications, such as image blending.

2. APPROXIMATE 4-2 COMPRESSORS

The number of partial products in the multiplication process can be reduced using the compressors. The compressor size can generally be represented as m:n, where m represents the number of input bits and n represents the number of output bits. As seen in Figure 2.7, an exact 4-2 compressor comprises two Full adders. It has three outputs and five inputs.

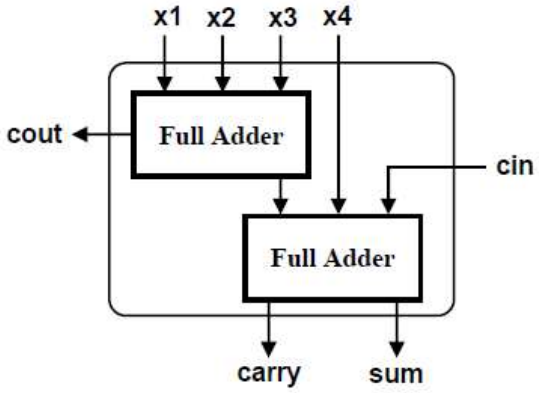

Figure 2.8. Exact 4:2 Compressor

Here, we design and analyze four Low-power approximate compressors C1, C2, C3, and C4.

C1 Compressor: In the C1 compressor, the sum is computed using majority logic for three bits, ignoring the fourth bit to reduce the complexity and Carry is simplified always to reflect the presence of a high input. The sum and carry equations for the C1 compressor are:

sum = (x1 x2) + (x2x3) +(x3x1) cy = x1+x2+x3+x4

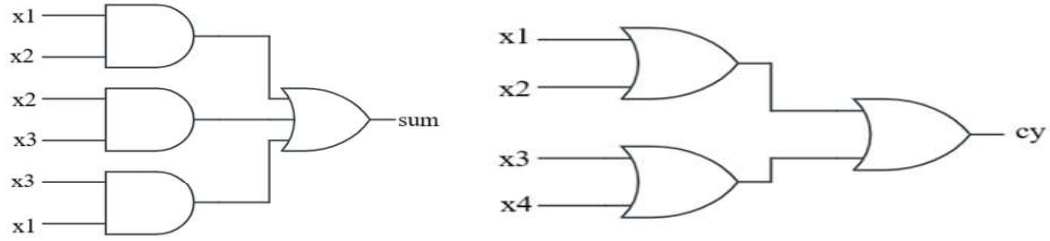

Figure 2.2. C1 Compressor

C2 Compressor: In the C2 Compressor, the sum is calculated as pairwise EX OR between the inputs, which reduces XOR computation complexity by splitting inputs into two groups. Carry relies on a simplified pairwise approach, introducing slight errors. The sum and carry equations for the C2 compressor are :

sum = (x1 ⊕ x2) + (x3 ⊕ x4) cy = (x1x2)+(x3x4)

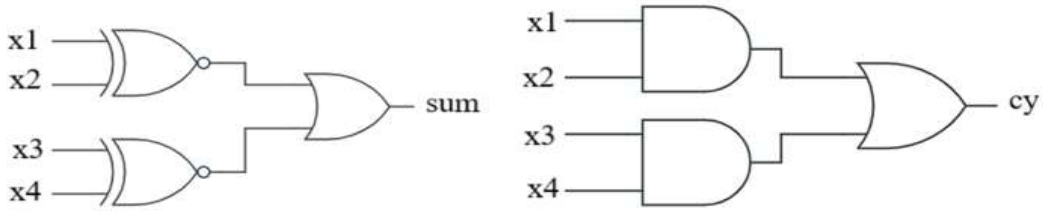

Figure 2.3. C2 Compressor

C3 Compressor: In the C3 Compressor, the sum and carry are calculated with three inputs only. Ignoring the fourth input in sum and carry simplifies the design. The sum and carry equations for the C3 compressor are :

sum = (x1 ⊕ x2 ⊕ x3) cy = (x1x2)+(x3)

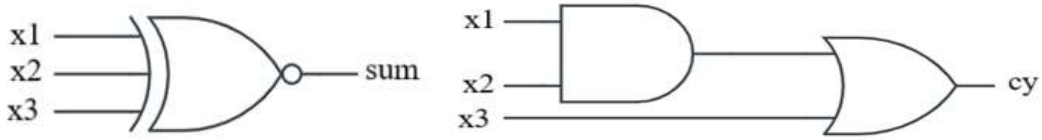

Figure 2.4. C3 Compressor

C4 Compressor: C4 Compressor Provides partial error compensation, balancing error reduction with added complexity. The sum and carry equations for the C4 compressor are :

sum = (x1 ⊕ x2 ⊕ x3) + x4 cy = (x1x2)+(x2x3)+(x3x4)

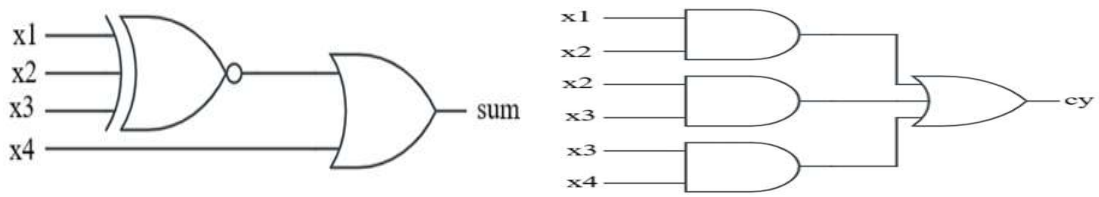

Figure 2.5. C3 Compressor

The Truth table in Table I compares the Exact 4:2 Compressor with four approximate compressor designs (C1, C2, C3, C4). The deviations in sum and carry values for approximate compressors indicate the level of inaccuracy introduced to optimize factors like power, area, and delay. Low-deviation compressors C1, and C2 offer better accuracy, suitable for moderate approximation applications, and Less power savings compared to C3 and C4 but still better than the exact design. The High-Deviation Compressors C3 and C4 offer Greater efficiency in power, delay, and area but at the cost of accuracy. These are Best for error-tolerant applications like image processing, AI, or approximate multipliers.

3. APPROXIMATE MULTIPLIER DESIGN

Multipliers are essential elements in digital signal processing systems, responsible for efficiently computing the product of two operands. In this section, we delve into the architecture of the proposed multipliers, focusing on their design principles, area utilization, propagation delay, and essential performance metrics like Peak signal-to-noise ratio (PSNR). Utilizing the Wallace tree multiplier method enables efficient computation of partial products and subsequent accumulation of results within the proposed architecture. The Wallace tree multiplier reduces the overall computation time while minimizing hardware resources by decomposing the multiplication operation into smaller segments and employing parallel processing. The hierarchical structure of the tree enables scalable implementation, making it suitable for a wide range of multiplication scenarios. The integration of custom compressors enhances the performance of the Wallace tree multiplier by optimizing area utilization and reducing delay. The proposed multipliers leverage a Wallace Tree architecture, a well-established method for reducing partial products and aggregating them to obtain the final product efficiently. The architecture shown in Figure 3.1 employs custom-designed compressors and full and half adders to optimize area utilization and minimize propagation delay. Each multiplier variant (PM1, PM2, PM3, PM4) employs a different compressor (C1, C2, C3, C4) to suit various application requirements.

Table 1 Truth Table of approximate 4:2 compressors

Inputs				Exact 4:2 Compressor		C1		C2		C3		C4	
x1	x2	x3	x4	sum	cy	sum	cy	sum	cy	sum	cy	sum	cy
0	0	0	0	0	0	0✓	0✓	0✓	0✓	0✓	0✓	0✓	0✓
0	0	0	1	1	0	0	1	1✓	0✓	0	0✓	1✓	0✓
0	0	1	0	1	0	0	1	1✓	0✓	1✓	1	1✓	0✓
0	0	1	1	0	1	1	1✓	1	0	1	1✓	1	1✓
0	1	0	0	1	0	0	1	1✓	0✓	1✓	0✓	1✓	0✓
0	1	0	1	0	1	1	1✓	1	0	1	0	1	1✓
0	1	1	0	0	1	1	1✓	1	1✓	0✓	1✓	1	1✓
0	1	1	1	1	1	1✓	1✓	1✓	1✓	0	1✓	1✓	1✓
1	0	0	0	1	0	0	1	1✓	0✓	1✓	0✓	1✓	0✓
1	0	0	1	0	1	1	1✓	1	0	1	0	1	1✓
1	0	1	0	0	1	1	1✓	1	1✓	0✓	1✓	1	1✓
1	0	1	1	1	1	1✓	1✓	1✓	1✓	0	1✓	1✓	1✓
1	1	0	0	0	1	1	1✓	1	1✓	1	1✓	1	1✓
1	1	0	1	1	1	1✓	1✓	1✓	1✓	1✓	1✓	1✓	1✓
1	1	1	0	1	1	1✓	1✓	1✓	1✓	0	1✓	1✓	1✓
1	1	1	1	1	1	1✓	1✓	1✓	1✓	0	1✓	1✓	1✓

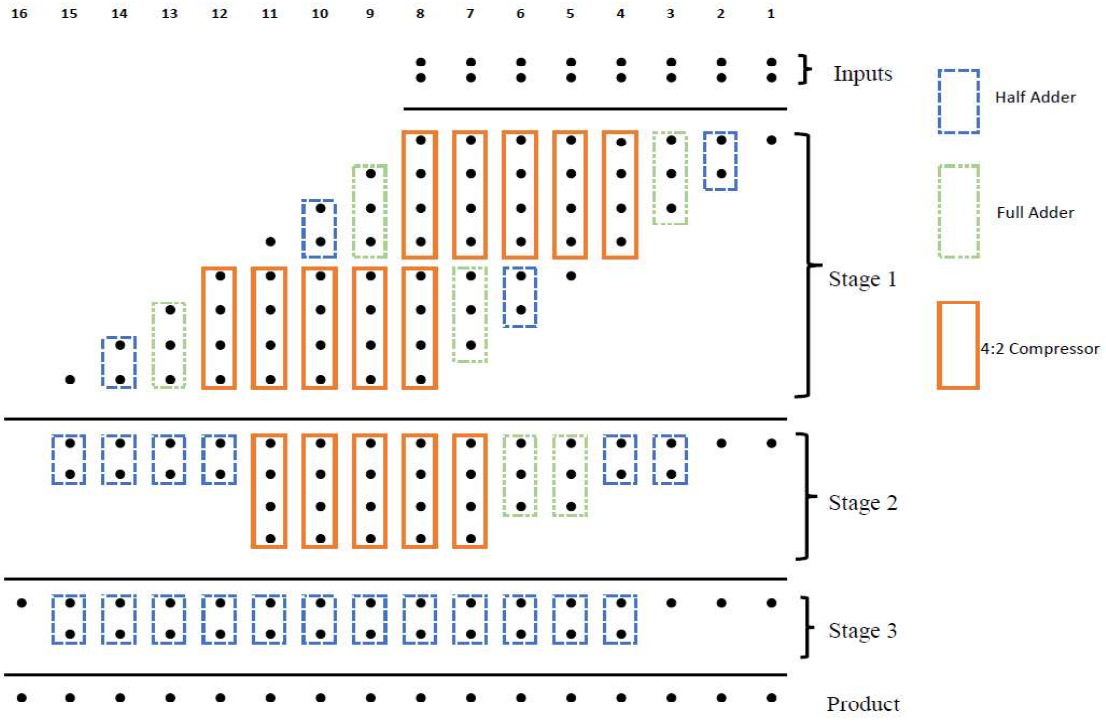

Figure 3.1. C3 Compressor Proposed Approximate 8x8 Multiplier

Multipliers designed using compressors such as C1, C2, C3, and C4 typically aim to optimize the speed and area of the multiplication process while managing complexity and error levels. Here's a brief overview of how each compressor might contribute to the design of multipliers:

1. **C1 Compressor**: The C1 compressor uses majority logic for summation, which helps in handling overflow and carries efficiently by simplifying the carry-out logic, allowing for faster computations. This design can be advantageous in a multiplier configuration, reducing the overall delay in calculating intermediate products.

2. **C2 Compressor**: By employing pairwise EX OR operations, the C2 compressor reduces the complexity of XOR calculations by grouping inputs. In multiplier designs, this approach can streamline the addition of partial products, making the overall multiplication process quicker. However, slight errors introduced in the carry calculations must be considered, as they can accumulate in larger multipliers.

3. **C3 Compressor**: The C3 compressor simplifies design by using only three inputs for sum and carry calculations, omitting a fourth bit. This can lead to reduced circuit complexity in multiplier layouts. However, careful handling of carry propagation is necessary, as the omission may introduce some inaccuracies in high-speed multiplication scenarios.

4. **C4 Compressor**: This compressor balances error reduction with added complexity by providing partial error compensation. In multipliers, the C5 compressor can help mitigate the negative effects of approximation, allowing for high-speed operations while maintaining a degree of accuracy in the multiplication results.

Each compressor offers distinct benefits and trade-offs that influence multiplier design. Choosing the appropriate compressor depends on the specific requirements of speed, area, and acceptable error levels in digital multipliers, with applications ranging from low-power designs to high-performance computing systems. Integrating these compressors can lead to innovative multiplier architectures that efficiently manage the challenges associated with multiplication in digital circuits, especially in contexts like image processing and digital signal processing.

4. PERFORMANCE EVALUATION

4.1 Accuracy Analysis:

To evaluate the accuracy of the designed multipliers, we compare their outputs with the exact results by calculating the Mean Relative Error Distance (MRED). The process begins by determining the Error Distance (ED), defined as $ED=|M'-M|$, where M' is the result produced by the approximate multiplier, and M is the exact result. Next, we calculate the Relative Error Distance (RED), given by $RED=ED/M$, representing the normalized error for each input combination. The MRED is then obtained as the average of all RED values across all possible input combinations:

$$MRED=\Sigma RED/65536$$

We perform an exhaustive simulation using MATLAB to evaluate all possible input combinations for each multiplier design. The results of this analysis are as follows:

Table 2: Accuracy comparison of Approximate 8x8 Multiplier

Design Name	ER (%)	MRED	NMED
MD -1 [14]	25.29	0.00228	3.50765×10^{-8}
PM1	36.27	0.01789	2.87524x10-7
MD-3 [14]	52.67	0.03066	4.71642×10^{-7}
MD-4 [14]	60.29	0.11778	1.81145×10^{-6}
PM2	74.23	0.14864	$3.52473\text{x}10^{-6}$
MD-5 [14]	92.53	0.21566	3.31665×10^{-6}
MD-7 [14]	95.27	0.23826	3.66419×10^{-6}
PM3	95.86	0.26897	$4.02567\text{x}10^{-6}$
MD-8 [14]	96.02	0.32955	5.06818×10^{-6}
PM4	97.73	0.28765	$4.28676\text{x}10^{-6}$

PM1 has a slightly higher error rate but still maintains a low MRED and NMED, making it a good balance between accuracy and approximation. PM3, and PM4 have the highest error rates and significantly higher MRED/NMED values, meaning they introduce a large amount of approximation error. Both PM2 and PM4 reduce power consumption by over 30% and improve energy efficiency by around 40%, making them excellent choices for low-power applications. PM4 is the best choice if the lowest PDP and smallest area are the main priorities. PM2 still offers good power savings with a slight trade-off in efficiency compared to PM4

4.2 Hardware Analysis:

To analyze the hardware performance of proposed multipliers, all multipliers are implemented in Verilog HDL and synthesized using Xilinx Vivado. For comparison purposes, an 8-bit exact Wallace tree multiplier was also implemented. The Exact design has the highest delay and power, making it the least power-efficient with the highest PDP. This suggests it is the least optimized among the designs. The PM4 design has the lowest PDP, indicating the best energy efficiency. It also has one of the lowest power consumption and a very low delay. PM2 is a strong contender with less PDP, a relatively low delay, and low power consumption. Both PM2 and PM4 significantly outperform the Exact multiplier in all metrics. PM4 is slightly better than PM2, with an additional 1-2% improvement in power, area, and PDP. PM2 and PM4 reduce power consumption by over 30% and improve energy efficiency by around 40%, making them excellent choices for low-power applications. PM4 is the best choice if the lowest PDP and smallest area are the main priorities. PM2 still offers good power savings with a slight trade-off in efficiency compared to PM4.

Table 3: Hardware Performance Comparison of Approximate 8x8 Multiplier

Design Name	Dealy (ns)	Power (µw)	Area(μm^2)	PDP(fJ)
Exact	7.938	7.588	758.56	60.24
MD -1 [14]	7.803	6.054	597.03	47.24
PM1	7.801	5.867	522.33	45.77
MD-3 [14]	7.803	5.645	512.35	44.05
MD-4 [14]	7.803	5.554	522.36	43.34
PM2	6.984	5.237	527.05	36.58
MD-5 [14]	6.974	5.974	534.26	41.67
MD-7 [14]	6.974	5.524	547.56	38.53
PM3	6.965	5.345	512.70	37.23
MD-8 [14]	6.972	5.214	514.65	36.36
PM4	6.964	5.148	512.47	35.86

5. IMAGE BLENDING APPLICATION

This work presents a novel image-blending technique implemented within the MATLAB environment. It leverages area-efficient approximate multipliers (M1, M2, M3, M4) to achieve finer control over the blending process compared to traditional methods. These custom multipliers prioritize specific image features or regions (e.g., edges in the cameraman image, textures in the Saturn image) during blending, offering a more nuanced approach while maintaining computational efficiency. The technique involves generating weight maps that define the contribution of each pixel from the source images (provided in TIF format, such as the classic cameraman and Saturn images) to the final blended output. The custom approximate multipliers then manipulate these weight maps to achieve the desired visual effect, such as the seamless blending of high-frequency details from the cameraman's image with the low-frequency textures of the Saturn image shown in fig 6. . The utilization of MATLAB's image processing toolbox streamlines the implementation process, enabling the creation and utilization of the multipliers. To evaluate the blended image's quality, we utilize established metrics like Peak signal-to-noise ratio (PSNR). The blended images are shown in Fig 7. These metrics offer quantitative assessments of the blended image's fidelity and accuracy, enabling thorough evaluation and analysis. These metrics provide quantitative measures of the blended image's fidelity compared to the original source images. Future research directions could involve exploring advanced, area-efficient multiplier designs that further optimize computational efficiency. Additionally, the integration of machine learning techniques holds promise for creating adaptive multipliers that can automatically adjust based on the specific content of the input images. Finally, developing a framework suitable for real-time image blending applications could expand the practical applications of this technique.

Figure 5.1. Cameraman and Saturn Inputs

Figure 5.2. Image Blending of different multipliers

5.1 Performance Evaluation

The performance of each multiplier variant is evaluated based on several key metrics, including area utilization, propagation delay, and PSNR for image blending applications.

Table 4 PSNR Comparison

Multiplier	PSNR
PM1	21
PM2	19
PM3	33
PM4	19

6.CONCLUSION

In this manuscript, we introduced a novel design for an approximate multiplier focusing on optimizing area utilization. We leverage compact compressors, half adders, and full adders to achieve this goal. The practical utility of our multiplier is demonstrated in an image blending application, under-scoring its relevance in real-world scenarios. Our thorough experimentation yields promising outcomes, with the highest peak signal-to-noise ratio (PSNR) reaching 33 for images of the cameraman and Saturn. Furthermore, we quantify the mean squared error (MSE) at 27, indicating the efficacy of our multiplier in maintaining image quality while reducing computational complexity. Our exhaustive evaluation under-scores the effectiveness of our approach in striking a balance between area efficiency and performance, rendering it suitable for deployment in resource-constrained settings like field-programmable gate arrays (FPGAs). The successful realization of our design underscores its potential across various domains requiring efficient arithmetic operations, thus advancing digital circuit design methodologies. Future research directions may explore further optimizations and extensions to address additional performance metrics and application domains.

7. REFERENCES

[1] Y. Wang, J. Lin, and Zh. Wang, “An Energy-Efficient Architecture for Binary Weight Convolutional Neural Networks,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 2, pp. 280-293, Feb. 2018.

[2] N. Zhu,W. L. Goh,W. Zhang, K. S. Yeo, and Z. H. Kong, “Design of Low- Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 8, pp. 1225-1229, Aug. 2010.

[3] Sr. Narayanamoorthy, H. A. Moghaddam, Zh. Liu, T. Park, and N. S.Kim, “Energy-Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180-1184, Jun. 2015.

[4] Ch. K. Jha, S. N. Ved, Is. Anand, and J. Mekie, “Energy and Error Analysis Framework for Approximate Computing in Mobile Applications,” IEEE Trans. Circuits Syst. II, Exp. Briefs. DOI: 10.1109/TCSII.2019.2910137.

[5] 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.

[6] Y. Liu, T. Zhang, and K. K. Parhi, “Computation Error Analysis in Digital Signal Processing SystemsWith Overscaled Supply Voltage,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 4, pp. 517-526, Apr. 2010.

[7] D. Jeon, M. Seok, Zh. Zhang, D. Blaauw, and D. Sylvester, “Design Methodology for Voltage-Overscaled Ultra-Low-Power Systems,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 59, no. 12, pp. 952-956, Dec. 2012.

[8] R. Liu and K. K. Parhi, "Power Reduction in Frequency-Selective FIR Filters Under Voltage Overscaling," IEEE Trans. Emerg. Sel. Topics Circuits Syst., vol. 1, no. 3, pp. 343-356, Sept. 2011.

[9] V. Mrazek, R. Hrbacek, Z. Vasicek and L. Sekanina, "EvoApprox8b:Library of Approximate Adders and Multipliers for Circuit Design and Benchmarking of Approximation Methods," in Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, Switzerland, Mar. 2017, pp. 258-261.

[10] A. Dalloo, A. Najafi , and A. G.-Ortiz, "Systematic Design of an Approximate Adder: The Optimized Lower Part Constant-OR Adder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 8, pp. 1595-1599, Aug. 2018.

[11] S. Dutt, S. Dash, S. Nandi, and G. Trivedi, "Analysis, Modeling and Optimization of Equal Segment Based Approximate Adders," IEEE Trans. Comput., vol. 68, no. 3, pp. 314-300, Mar. 2019.

[12] O. Akbari, M. Kamal, A. A.-Kusha, and M. Pedram, "RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 65, no. 8, pp. 1089-1093, Aug. 2018.

[13]M. Ha and S. Lee, "Multipliers with Approximate 4-2 Compressors and Error Recovery Modules," IEEE Embedded Systems Letters, vol. 10 , no. 1, pp. 6-9, Mar. 2018.

[14] M.S. Ansari, H. Jiang, B.F. Cockburn, J. Han, Low-power approximate multipliers using encoded partial products and approximate compressors, IEEE J. Emerging Sel. Top. Circuits Syst. 8 (3) (2018) 404–416, https://doi.org/10.1109/ JETCAS.2018.2832204. [2] A. Jain, N. Dugar, S. Yadav

Biographies

Sreelakshmi. Vadlamudi is a research scholar in the Electronics & Communication Engineering department under the JNTUK, Kakinada, Andhra Pradesh. She is currently working as an Assistant Professor in, the Electronics & Communication Engineering department in Seshadri Rao Gudlavalleru Engineering College. She completed UG in Electronics and Communication Engineering in the year 2004 from JNTU, Hyderabad. She completed her PG in VLSI System Design in 2011 at JNTU, Hyderabad.

Dr.Y. Syamala received her B.E., M.E., from Bharathiyar University, Anna University in 2001, and 2005 respectively. She received a Ph.D from JNTUH; Hyderabad in 2014. She has been working as a Professor at Seshdri Rao Gudlavalleru Engineering College, Gudlavalleru. Her research interest includes Low-power VLSI design, Digital design, and Testing.

Madhu Ramarakula received the BE (Bachelor of Engineering) degree in electronics and communication engineering from Osmania University, Hyderabad, India, in 2003 and MTech degree in communication systems from Jawaharlal Nehru Technological University Hyderabad, India, in 2009. He received his PhD degree in electronics and communication engineering from Andhra University, Visakhapatnam, India, in 2014. He is presently working as an assistant professor in the Department of Electronics and Communication Engineering, University College of Engineering Kakinada (A), JNTUK, Kakinada, India. He has 10 years of teaching experience. He has published more than 40 research papers in various reputed national and international journals and conferences. His research interests include mobile and cellular communications, antennas, satellite communications, and GNSS. He is a Member of IEEE.