Design and Performance Analysis of Power-Efficient Multipliers for Image Sharpening

Tanya Mendez
Nitte (Deemed to be University), NMAM Institute of Technology (NMAMIT), Nitte, Department of Robotics and Artificial Intelligence, Udupi, India
tanya.mendez@nitte.edu.in (corresponding author)

Manjunatha Badiger
Nitte (Deemed to be university), NMAM Institute of Technology (NMAMIT), Nitte, Department of VLSI Design and Technology, Udupi, India
manjunatha.badiger@nitte.edu.in

Received: 8 July 2025 | Revised: 24 July 2025, 6 August 2025, and 17 August 2025 | Accepted: 22 August 2025

 DOI: https://doi.org/10.48084/etasr.13232

ABSTRACT

Strong motivation to reduce power consumption in portable devices with battery operation has driven research into novel power-efficient digital circuit approaches. Modern processors integrate complex multipliers, but increased computational complexity leads to longer delays and higher power consumption. The need for high-speed multipliers with low power consumption arises to achieve optimal performance of digital image and signal processing applications. The advent of emerging technologies is the driving force behind the design of multipliers with minimized power consumption, reduced delays, and smaller dimensions. This study presents a comparative analysis of various multiplier architectures and proposes two novel designs: (i) Low-Power Approximate Multiplier using the Nikhilam algorithm (LPAMN) and (ii) Efficient Carry-Save Multiplier with SBETA Optimization (ECSM-SBETA-O). These architectures leverage approximate computing techniques to minimize power consumption while maintaining computational accuracy. The proposed designs were implemented using the Cadence Genus tool with the GPDK-90 nm and GPDK-45 nm technology libraries. LPAMN achieved a significant 31.59% power reduction, along with 40.03% and 47.43% improvements in PDP and EDP, respectively, compared to existing designs. Furthermore, it enhanced image sharpening performance, achieving an improved PSNR of 32.16 dB. These results highlight the potential of the proposed designs for energy-efficient digital processing applications.

Keywords-low-power; multiplier; Verilog; approximate computing; Vedic arithmetic; image processing

I. INTRODUCTION

A significant contribution to the design and development of low-power multipliers is evident, since the CPUs of most high-speed applications require them as one of the crucial components. Moore's law asserts that the number of circuits on a chip doubles every 18 months, increasing power dissipation due to the soaring number of circuits. The growing market demand for battery-operated, lightweight, and small digital electronics is also drawing attention to research into low-power devices. To cope with these increasing demands for high performance, circuits per chip, power consumed, and clock frequencies are growing. Power constraints have restricted the frequency to approximately 4.5 GHz, which is insufficient for the high-performance requirements of high-speed devices.

Frequencies can be increased by reducing the power consumed at a trade-off between increased delay and the high-frequency demands of high-performance devices. The demand for innovative techniques on low-power VLSI is driven mainly by high-speed circuits designed to operate at higher clock frequencies to meet the needs of high-performance processors. Various approaches have been explored to meet high-performance demands, but this study focuses on approximate computing and ASIC-specific optimizations [1, 2].

The rapid evolution of VLSI technology is driven by innovations that focus on making microchips more efficient, cheaper, faster, and smaller. This has improved chip density and reduced power consumption for high-performance devices. The growing demands of contemporary technologies, such as AI, 5G, and IoT, are driven by innovations in VLSI technology. Integrated Circuits (ICs) are the basic building blocks of VLSI, using millions of transistors on a single chip. VLSI is the backbone of contemporary electronics, enabling the development of complex circuits in electronic devices. Enhanced performance, reduced power consumption, and miniaturization of digital devices have become a reality due to the integration of various functions into compact chips using innovative VLSI techniques.

Approaches to reduce power and enhance efficiency have become more widespread in recent decades [3, 4]. These approaches can be applied at various levels of abstraction, such as application, system, architecture, circuit, and technology. The design hierarchy should incorporate optimization for low power. Such techniques reduce switching activity, supply voltage, capacitance, and voltage to minimize overall power dissipation. Reduction in power consumption can be achieved by upgrading the components of computationally demanding parts with application-specific logic [5]. However, system performance can be improved by sacrificing computational accuracy using the concepts of approximate arithmetic computing at the application level [6, 7].

The increasing demand for low-power devices and the advancement of VLSI mainly trigger the increasing utilization of low-power design approaches [8]. Significant benefits have been observed in the design flow as a result of incorporating low-power techniques and optimization tools. The power consumption of a system can be minimized by incorporating suitable low-power techniques, depending on the abstraction level. When low-power techniques are applied at higher levels of abstraction, the power savings are more significant [9]. In addition, considerable power savings can be achieved using low-power techniques at the architectural and algorithmic levels. At higher levels of integration, optimizations at the transistor or logic level do not satisfy low power requirements, emphasizing the research focus on power optimization at higher design flow levels.

An analysis of different multiplier types was carried out, using the concepts of approximate computing and Vedic mathematics to identify the ideal multiplier for image processing applications. In [10], a Vedic binary multiplier incorporated the concept of Nikhilam sutra. Significant speed improvements were achieved by adapting the multiplication of larger to smaller operands with an addition operation. An enhanced efficiency was observed for higher-order arithmetic operations, as fewer clock cycles were required for the proposed algorithm. In [11], the Yavadunam algorithm of Vedic mathematics was utilized in the design of a binary multiplier. Multiplication of a wide range of binary numbers was possible using the proposed multiplier with reduced layout complexity and substantial speed improvements. Power savings were observed in multipliers designed for higher-order bits.

In [12], the proposed multiplier was based on the concept of Nikhilam sutra, being suitable for complex numbers. The multiplier was designed using an algorithm that consisted of selecting a radix, determining exponents, and multiplying. Incorporating parallelism for the multiplication operation improved the speed due to the reduction in the carry propagation delay, resulting in superior performance. In [13], efficient, low-cost, approximate multipliers were developed for error-tolerant digital image processing applications, reducing hardware complexity along with accuracy. MATLAB simulations showed that the proposed multipliers exhibited performance on par with existing designs, demonstrating their capabilities for image multiplication operations in terms of SNR and PSNR.

SAFAN [14] stands for Serial Approximate Full Adder using NAND gates, using Material Implication (IMPLY) for in-memory computing applications. The widespread requirement for power-saving computational processes has generated memristor-based devices as promising dual-operation components for computational processing and data storage. In [15], a novel left-to-right binary adder architecture was inspired by the Vedic addition method, targeting improvements in the VLSI area and power efficiency. By initiating the addition from the Most Significant Bit (MSB), where carry generation is minimal, the proposed design effectively reduces both area and power-delay metrics. Overall, the architecture offers a promising solution for computationally intensive VLSI applications that require optimized speed and resource utilization.

This work investigates different low-power multiplier designs and proposes two low-power multipliers incorporating approximate adders, one based on the Nikhilam algorithm (LPAMN) and the other based on the Urdhva Tiryagbhyam (UT) sutra algorithm (ECSM-SBETA-O).

II. PROPOSED LOW-POWER MULTIPLIER ARCHITECTURES

This study explored the algorithms of Vedic mathematics, particularly Nikhilam sutra and UT sutra, along with the low-power VLSI concept of approximate computing, to design two multiplier architectures using ASIC design. The Vedic arithmetic approach allows for the conception of fast, low-power multipliers. The proposed design implements a multiplier with reduced power requirements relative to standard ones. Approximate computing is a design paradigm that trades computational accuracy for performance, power efficiency, and resource utilization improvements. By allowing controlled approximations in software or hardware, approximate computing reduces energy consumption and enhances processing speed, making it ideal for resource-intensive tasks. This approach is widely applied in VLSI design, low-power FPGA development, and deep-learning circuit implementations to achieve efficient computing solutions. A comparative analysis of the proposed and existing 8-bit and 16-bit multiplier architectures, with respect to power, area, delay, PDP, and EDP, was performed using the Cadence Genus tool, synthesized using 90- and 45-nm standard technology libraries.

A. Low-Power Approximate Multiplier Using the Nikhilam Algorithm (LPAMN)

This study proposes a new multiplier design based on Vedic mathematics to achieve low power and high speed performance. The computational speed of the multiplier is increased, as the operations based on Vedic mathematics require less hardware and are quite fast. Nikhilam sutra from Vedic mathematics for multiplication was explored to design the multiplier. Nikhilam Navatas' Charamam Dasatah is a Vedic sutra that means "all from 9 and last from 10". Large digit multiplication can be converted to small digit multiplication using a few add, subtract, and shift operations, reducing carry propagation delay. An added advantage of Nikhilam sutra is that it can reduce the number of steps involved in multiplication [16]. Power consumption is further

reduced through the incorporation of the Selector-Based Error Tolerant Adder (SBETA) that was implemented in [17], utilizing the concept of approximate computing. Figure 1 illustrates the implementation of the proposed LPAMN.

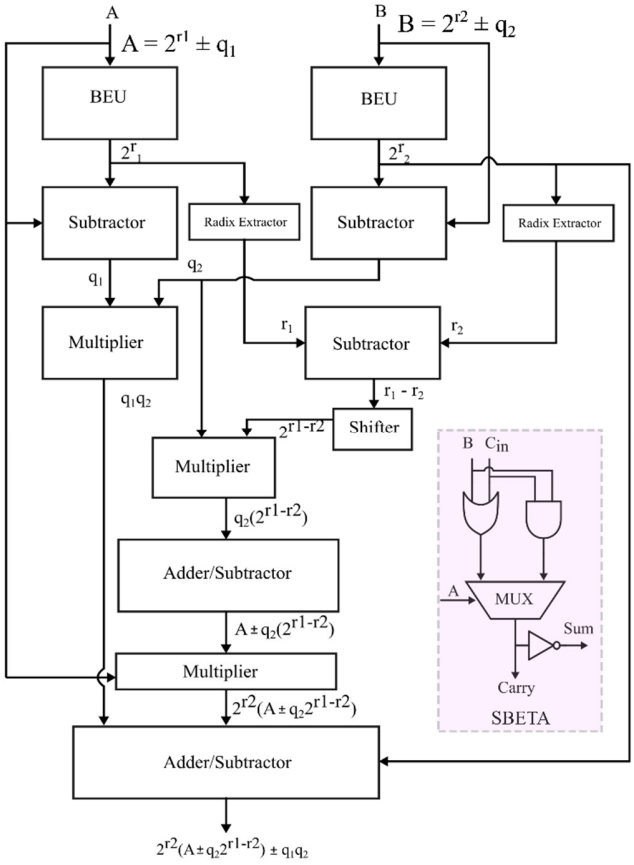

Fig. 1. LPAMN design.

The SBETA performs the arithmetic operations, mainly addition and subtraction, within the multiplier, permitting meticulous approximation with minimal impact on accuracy, reducing computation time and saving power. In the SBETA, the adder is divided into two parts: The accurate block, where the accurate computation of the MSBs is performed, ensuring minimal error in the final output, and the approximate block, where the computation of the Least Significant Bits (LSBs) is performed using approximate computing.

The various add/subtract operations that are required after the base-based transformations in the Nikhilam multiplier are replaced with the SBETA, making it power-efficient without significantly degrading the final product. This is useful for applications where 100% accuracy is not required, such as in image processing applications. The proposed LPAMN design effectively combines the Nikhilam sutra for computational simplicity with the SBETA for power efficiency. This hybrid approach optimizes both speed and energy, making it robust for approximate arithmetic units in modern VLSI systems. This multiplier is termed approximate because it introduces controlled error through the SBETA units used in the addition/subtraction stages of the architecture.

Two operands, A and B, of n bits, are assumed for the multiplication operation. These two n -bit numbers are considered with radices r_1 and r_2 . The mathematical expressions for the proposed multiplier are given by:

$$A = 2^{r_1} \pm q_1 \quad (1)$$

$$B = 2^{r_2} \pm q_2 \quad (2)$$

$$A * B = 2^{r_2}(A \pm q_2 2^{k_1 - k_2}) \pm q_1 q_2 \quad (3)$$

1) Base Determinant Unit

A base determinant unit is required to determine the base/radices related to the input numbers, as depicted in Figure 2. A priority encoder is used to extract the base or radix of the number. The base of the input number $p1$ is given to a shifter to obtain 2^{p1}. The base is incremented by 1 $(p1 + 1)$ and given to another shifter to obtain 2^{p1+1}. The mean of 2^{p1} and 2^{p1+1} is computed to decide the necessary base/radix. The data input is compared with the computed mean in the comparator block. When the data input is greater than the computed mean, $p1 + 1$ is chosen as the base $(r1)$. When the data input is less than the computed mean, $p1$ is chosen as the base $(r1)$. The output of the base determinant unit is the selected radix 2^{r1}

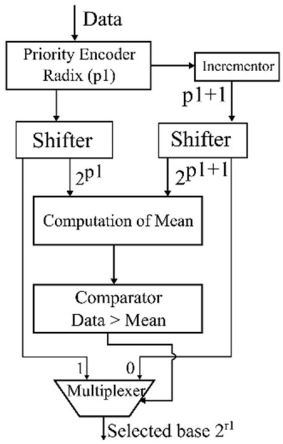

Fig. 2. Base determinant block diagram.

The output obtained from the base determinant unit has the form 2^{r1} and 2^{r2}, which is given to a subtractor to obtain the remaining portion q_1 and q_2 after subtracting 2^{r1} and 2^{r2} from the data input A and B, respectively. A multiplier is then used to obtain $q_1 * q_2$. A radix extractor then extracts $r1$ and $r2$, which are fed to a subtractor module to obtain $r1 - r2$.

The output of the subtractor $r1 - r2$ is given to a shifter to obtain 2^{r1-r2}. A multiplier is then used, which takes as input 2^{r1-r2} and q_2 to obtain $q_2\ 2^{r1-r2}$. This output is given to an adder/subtractor along with the first data input (A) to obtain $A \pm q_2\ 2^{r1-r2}$. This adder/subtractor output $(A \pm q_2\ 2^{r1-r2})$, along with the shifter output 2^{r2}, are given to a multiplier to obtain $2^{r2}(A \pm (q_2)2^{r1-r2}$. A final adder/subtractor block is used to compute $2^{r2}(A \pm (q_2)2^{r1-r2} \pm q_1 q_2$.

B. Efficient Carry-Save Multiplier with SBETA Optimization (ECSM-SBETA-O)

Arithmetic multiplication is one of the most fundamental operations in all processing units. The CPU performance can increase through improved PDP optimization of multipliers. This work explores the design of an iterative carry-save multiplier through implementation with SBETA, designed in [18], which uses the low-power concept of approximate computing. Partial product addition through Vedic arithmetic increases the speed of multiplication operations without the necessity of separate hardware. The initial multiplication stage begins with the generation of partial products achieved using the logic AND operation.

The second stage begins with designing an 8×8 multiplier by implementing iterative carry save addition and SBETA. SBETA controls the modular implementation of this process, which utilizes full and half adders combined with the SBETA design, as depicted in Figure 3. Each column's summation between partial products produces both final product bits and carry results. The CLA executes the final calculation for the result. The implementation of iterative carry-save multipliers using SBETA results in reduced computational duration compared to the RCA-based multiplier.

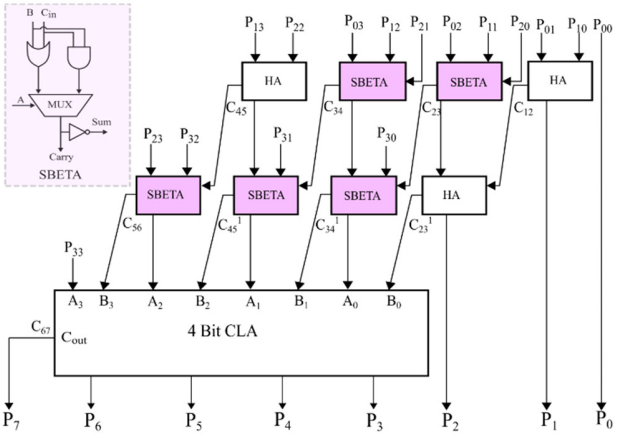

Fig. 3. SBETA-based 8×8 iterative multiplier with carry-save addition.

The execution time of an efficient carry-save multiplier with SBETA optimization is given by,

$$t_{ECS} = (n-2)t_{SBETA} + t_{CLA} \quad (4)$$

where the multiplier implemented using RCA has a summation time given by

$$t_{RCA} = 2(n-1)t_{FA} + t_{RC} \quad (5)$$

where the number of rows is n and the number of columns is $2n-1$.

The third step employs the UT sutra to execute 16-bit multiplication through four SBETA-based 8×8 iterative multipliers with carry-save addition, as presented in Figure 4.

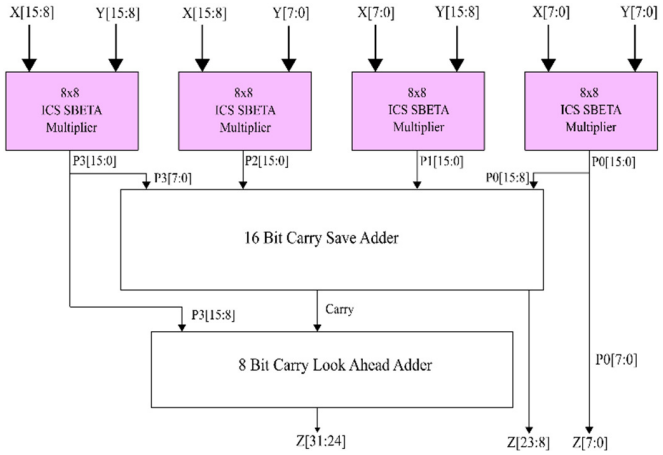

Fig. 4. SBETA-based 16×16 iterative multiplier with carry-save addition.

Devices using iterative carry-save multipliers reach operation speeds that are twice as fast as those based on the RCA method present in conventional multipliers. The addition process reaches higher speeds through the implementation of CLA during the final stage.

III. RESULTS AND DISCUSSIONS

The ASIC design was carried out on all 16-bit multiplier designs mentioned in this work. The multipliers were synthesized using the Cadence Genus tool, incorporating standard 45- and 90-nm GPDK libraries. The comparative analysis of the proposed and existing 16-bit multipliers' performance is analyzed in terms of different metrics, as shown in Table I. The same framework simulates and synthesizes the proposed and existing multiplier designs. The RTL design entry uses the Verilog Hardware Description Language (VHDL) to generate code that the synthesis process maps to library cells. Random binary inputs are used to compare actual products with predicted products to validate the functionality of both proposed and literature-based multipliers. The Cadence Incisive tool performs RTL verification tasks, varying input toggle rates to perform power analysis. The logic synthesis process transforms RTL into netlist information using the process library. The functional definition of the design comes from the RTL documents, while the process library gives essential characteristics to logic gates. The Electronic Design Automation (EDA) tool performs optimal transformation, and this work uses Cadence Genus together with 90- and 45-nm technology libraries for the logic synthesis of the proposed low-power multipliers and existing designs. The proposed 16-bit and existing multipliers were evaluated for various performance metrics in terms of delay, area, power, Power-Delay Product (PDP), and Energy-Delay Product (EDP), as shown in Table I. Based on the results from Table I, the power analysis of the proposed and existing 16-bit multipliers revealed the following inferences. At 90 nm, the lowest power consumption of 192.65 µW was observed in the ECSM-SBETA-O, followed by the LPAMN that exhibited a power consumption of 198.63 µW. At 45 nm, the lowest power consumption was again exhibited by the ECSM-SBETA-O with a value of 866.92 µW, while the Yavadunam Sutra-based Multiplier consumed the highest power of 2,919 µW.

TABLE I. COMPARATIVE ANALYSIS OF PROPOSED AND EXISTING 16-BIT MULTIPLIERS

Techn. Lib.	90 nm					45 nm				
Architecture	**Power (µW)**	**Delay (nS)**	**Area (µm²)**	**PDP (fJ)**	**EDP (e-24 Js)**	**Power (µW)**	**Delay (nS)**	**Area (µm²)**	**PDP (fJ)**	**EDP (e-24Js)**
LPAMN	198.63	4.71	18342.1	935.54	4,406.42	893.83	6.31	18091.1	5640.1	35588.82
ECSM-SBETA-O	192.65	4.24	12214.6	816.83	3,463.38	866.92	5.6816	11953.7	4925.5	27984.68
Fast multiplier [10]	726.17	3.71	197984	2,694.1	9,995.2	820.57	4.9714	197721	4079.4	20280.24
Yavadunam Sutra-based Multiplier [11]	729.75	6.10	20160	4,451.5	27,153.9	2,919.0	8.174	19904	23859	195030.9
Nikhilam Algorithm Multiplier [12]	218.85	4.97	21660.5	1,087.7	5,405.8	875.4	6.6598	21395.5	5829.9	38826.56
ICS-based Multiplier [19]	201.12	4.55	13415.7	915.09	4,163.7	905.04	6.097	13151.7	5518.0	33643.42
Vedic Pipeline [20]	243.16	5.70	24905.6	1,386.7	7,900.3	972.64	7.638	24642.6	7429.0	56742.89
High-speed Multiplier [21]	371.69	5.34	24131.2	1,984.9	10,598.9	1,486.76	7.1556	23878.2	10638	76125.99

PDP represents the trade-off between power and delay. At 90 nm, the most power-efficient multiplier was observed to be the ECSM-SBETA-O, with a PDP of 816.83 fJ. In contrast, the highest PDP of 4,451.47 fJ was achieved by the Yavadunam Sutra-based Multiplier, indicating a significant power-delay overhead. At 45 nm, the proposed ECSM-SBETA-O had the lowest PDP of 4925.5 fJ.

These results show that the proposed ECSM-SBETA-O and LPAMN were the most efficient designs in terms of power and energy efficiency in image processing applications.

A. Application of the Proposed Low-Power Multipliers on Image Sharpening

The evaluation of the proposed low-power multipliers was performed using image sharpening, using a convolution operation with a 3×3 Laplacian sharpening kernel. Each pixel is updated based on its neighboring pixel values, weighted by the kernel coefficients. Multiplication occurs between the kernel coefficients and the pixel values. Incorporating approximate computing in low-power multipliers reduces power, delay, and area while introducing minor errors, which are acceptable in image processing applications. The input image is first converted to grayscale format for simplified processing. Anisotropic diffusion, a smoothing technique that preserves edges and reduces noise, is then applied to enhance image quality.

The preprocessed image is resized to a compact, hardware-friendly 3×3 pixel format, making it suitable for FPGA-based processing tasks such as image sharpening or compression. A Verilog-based image sharpening module applies a 3×3 Laplacian kernel to enhance edges and details in an image. It begins by taking a 3×3 pixel input window with predefined hexadecimal pixel values. Each pixel value is multiplied by the corresponding kernel value using the proposed low-power multipliers that truncate less significant bits to save power and hardware resources. The results of these approximate multiplications are summed up, producing a sharpened pixel value. The output is normalized to ensure proper representation, and values exceeding the maximum allowable pixel intensity are clamped to this limit. The Verilog simulation clearly demonstrates this sharpening procedure in hardware, resulting in improved visual clarity by enhancing image edges and contrast in a computationally efficient way suitable for FPGA implementation.

Figure 5 depicts the original [22] and sharpened CT image obtained after image sharpening using the proposed LPAMN. The PSNR values of the proposed multiplier were 32.16 dB with a Laplacian filter and 33.78 dB with a strong Laplacian filter, compared to the accurate multiplier that gave a PSNR of 15.32 dB and 17.12 dB.

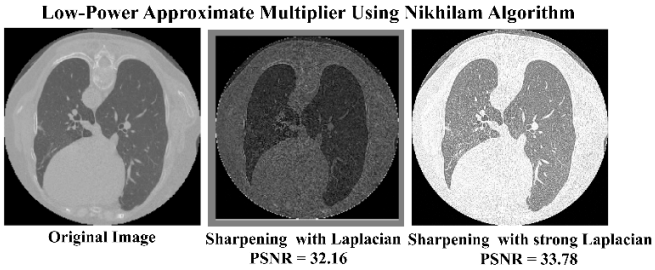

Fig. 5. Original and sharpened image obtained after image sharpening using the proposed multiplier.

IV. CONCLUSION

This paper presented the design of two low-power multiplier architectures, LPAMN and ECSM-SBETA-O, utilizing the concepts of approximate computing. The proposed and existing multiplier architectures were synthesized using VHDL with the Cadence Genus tool, incorporating the 90- and 45-nm standard GPDK libraries. A comparative analysis of the proposed and existing 16-bit multiplier architectures was performed with respect to power, area, delay, PDP, and EDP, and the results indicate that the proposed multipliers show significant benefits. The proposed LPAMN algorithm showed the best PDP of 935.54 fJ compared to other multiplier architectures and was utilized in an image sharpening application with an improved PSNR of 32.16 dB. The 16-bit LPAMN exhibited 9.23%, 13.98%, and 18.48% improvements, respectively, using 90-nm GPDK standard libraries in comparison to the Nikhilam algorithm [12]. In the future, LPAMN can be used for complex image and signal processing applications with tight power and delay constraints.

REFERENCES

[1] A. M. Dalloo, A. Jaleel Humaidi, A. K. Al Mhdawi, and H. Al-Raweshidy, "Approximate Computing: Concepts, Architectures, Challenges, Applications, and Future Directions," *IEEE Access*, vol. 12, pp. 146022–146088, 2024, https://doi.org/10.1109/ACCESS.2024.3467375.

[2] S. Mittal, "A Survey of Techniques for Approximate Computing," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–33, May 2016, https://doi.org/10.1145/2893356.

[3] S. Sorrell, "Reducing energy demand: A review of issues, challenges and approaches," *Renewable and Sustainable Energy Reviews*, vol. 47, pp. 74–82, Jul. 2015, https://doi.org/10.1016/j.rser.2015.03.002.

[4] M. R. Z. Banadkouki, "Selection of strategies to improve energy efficiency in industry: A hybrid approach using entropy weight method and fuzzy TOPSIS," *Energy*, vol. 279, Sep. 2023, Art. no. 128070, https://doi.org/10.1016/j.energy.2023.128070.

[5] A. Majeed and M. A. Shah, "Energy efficiency in big data complex systems: a comprehensive survey of modern energy saving techniques," *Complex Adaptive Systems Modeling*, vol. 3, no. 1, Dec. 2015, Art. no. 6, https://doi.org/10.1186/s40294-015-0012-5.

[6] P. Anguraj and T. Krishnan, "Design and realization of area-efficient approximate multiplier structures for image processing applications," *Microprocessors and Microsystems*, vol. 102, Oct. 2023, Art. no. 104925, https://doi.org/10.1016/j.micpro.2023.104925.

[7] M. Narayanappa and S. S. Yellampalli, "Approximate single precision floating point adder for low power applications," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 13, no. 3, Nov. 2024, Art. no. 650, https://doi.org/10.11591/ijres.v13.i3.pp650-664.

[8] T. Shan, "Advancements in VLSI low-power design: Strategies and optimization techniques," *Applied and Computational Engineering*, vol. 41, no. 1, pp. 22–28, Feb. 2024, https://doi.org/10.54254/2755-2721/41/20230706.

[9] T. J. Nagalakshmi, "An overview of low power technologies and the Alternative Approaches for low power IOT Architecture," in *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, India, Jan. 2024, pp. 1–6, https://doi.org/10.1109/ICAECT60202.2024.10468962.

[10] S. R. Sahu, B. K. Bhoi, and M. Pradhan, "Fast signed multiplier using Vedic Nikhilam algorithm," *IET Circuits, Devices & Systems*, vol. 14, no. 8, pp. 1160–1166, Nov. 2020, https://doi.org/10.1049/iet-cds.2019.0537.

[11] A. Deepa and C. N. Marimuthu, "Design of a high speed Vedic multiplier and square architecture based on Yavadunam Sutra," *Sādhanā*, vol. 44, no. 9, Sep. 2019, Art. no. 197, https://doi.org/10.1007/s12046-019-1180-3.

[12] P. Saha, A. Banerjee, A. Dandapat, and P. Bhattacharyya, "ASIC design of a high speed low power circuit for factorial calculation using ancient Vedic mathematics," *Microelectronics Journal*, vol. 42, no. 12, pp. 1343–1352, Dec. 2011, https://doi.org/10.1016/j.mejo.2011.09.001.

[13] B. Rashidi, "Efficient and low-cost approximate multipliers for image processing applications," *Integration*, vol. 94, Jan. 2024, Art. no. 102084, https://doi.org/10.1016/j.vlsi.2023.102084.

[14] S. Asgari, M. R. Reshadinezhad, and S. E. Fatemieh, "Energy-efficient and fast IMPLY-based approximate full adder applying NAND gates for image processing," *Computers and Electrical Engineering*, vol. 113, Jan. 2024, Art. no. 109053, https://doi.org/10.1016/j.compeleceng.2023.109053.

[15] M. G. Anuradha and R. Arun Kumar, "A Novel and Efficient Left-to-Right Binary Adder Architecture for reduced Area and Power Metrics in VLSI Design," *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 22629–22635, Jun. 2025, https://doi.org/10.48084/etasr.9840.

[16] C. Eyupoglu, "Investigation of the Performance of Nikhilam Multiplication Algorithm," *Procedia - Social and Behavioral Sciences*, vol. 195, pp. 1959–1965, Jul. 2015, https://doi.org/10.1016/j.sbspro.2015.06.208.

[17] T. Mendez, S. G. Nayak, V. P. Kumar, S. R. Vijay, and V. K. Kedlaya, "Performance Metric Evaluation of Error-Tolerant Adders for 2D Image Blending," *Electronics*, vol. 11, no. 15, Aug. 2022, Art. no. 2461, https://doi.org/10.3390/electronics11152461.

[18] T. Mendez and S. G. Nayak, "Design and Analysis of an Iterative Carry Save Adder-based Power-Efficient Multiplier," *Iranian Journal of Electrical and Electronic Engineering*, vol. 18, no. 1, Mar. 2022, https://doi.org/10.22068/IJEEE.18.1.2238.

[19] G. R. Gokhale and P. D. Bahirgonde, "Design of Vedic-multiplier using area-efficient Carry Select Adder," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, India, Aug. 2015, pp. 576–581, https://doi.org/10.1109/ICACCI.2015.7275671.

[20] A. Eshack and S. Krishnakumar, "Reversible logic in pipelined low power vedic multiplier," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 3, pp. 1265–1272, Dec. 2019, https://doi.org/10.11591/ijeecs.v16.i3.pp1265-1272.

[21] C. V. S. Chaitanya, C. Sundaresan, P. R. Venkateswaran, K. Prasad, and V. S. Ramakrishna, "Design of High-Speed Multiplier Architecture Based on Vedic Mathematics," *International Journal of Engineering & Technology*, vol. 7, no. 2.4, Mar. 2018, Art. no. 105, https://doi.org/10.14419/ijet.v7i2.4.11228.

[22] "Computed Tomography (CT) of the Brain." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/trainingdatapro/computed-tomography-ct-of-the-brain.