Artificial Intelligence and Electrical & Electronics Engineering: AIEEE Open Access ISSN: 3069-0706

PRIME OPEN ACCESS

Volume 1, Issue 2

Research Article

Date of Submission: 26 May, 2025
Date of Acceptance: 08 September, 2025
Date of Publication: 17 September, 2025

Exploring Shape and Margin in Analog Computing Cir-cuits: A Machine Learning-Based Approach to Design and Performance Evaluation

Abhishek Agwekar[1*] and Laxmi Singh[2]

[1]PhD. Scholar, Department of Electronics & Communication Engineering, Rabindranath Tagore University (RNTU), Bhopal

[2]Professor & Head, Department of Electronics & Communication Engineering, Rabindranath Tagore University (RNTU), Bhopal

***Corresponding Author:**
Abhishek Agwekar, PhD. Scholar, Department of Electronics & Communication En-gineering, Rabindranath Tagore University (RNTU), Bhopal.

Citation: Agwekar, A., Singh, L. (2025). Exploring Shape and Margin in Analog Computing Cir-cuits: A Machine Learning-Based Approach to Design and Performance Evaluation. *Art Intelligence and Ele & Electronics Eng: AIEEE Open Access, 1*(2), 01-08.

Abstract
This research focuses on the design and analysis of shape-based analog computing (S-AC) circuits utilizing the margin-propagation method. It examines the fundamental characteristics of S-AC circuits, particularly their scalability in terms of precision, speed, and power efficiency when compared to digital alternatives. The development of S-AC circuits inte-grates machine learning (ML) architectures with mathematical functions, and their input-output characteristics are modeled using a CMOS process for circuit simulations.

A key advantage of S-AC-based neural networks is their robustness against temperature variations, ensuring consistent accuracy. Additionally, as the accuracy of the fundamental S-AC process improves with multiple splines, scalability remains unaffected. This paper also highlights the significance of Design Margin and Shape Analysis, where the design parameter SSS and ML-based techniques are critical in shaping the system's ability to replicate the desired functional form. Unlike conventional design approaches, S-AC design provides flexibility by allowing users to define the proto-shape based on application-specific requirements, prioritizing the achievement of precise functional forms.

Keywords: Fabrication Scalability, Error Margin, Predictive Modelling, S-AC Computing

Introduction
In the Moderate Inversion (MI) regime, power and speed are always optimally balanced. The primary faults affecting the performance of S-AC circuits include performance mismatches, noise, and power-supply fluctuations. These undesirable factors introduce additive errors that can significantly impair circuit functionality. The variations caused by second-order effects are explained by the gap between the forms observed in the Strong Inversion (SI) and Weak Inversion (WI) regimes of S-AC circuits. According to P. Kumar and Chakraborty, this characteristic enables S-AC circuits to preserve the intrinsic form of the implemented function [1,2].

This study examines how nominal temperature variations affect S-AC units. The characteristic curves of S-AC-based ReLU, DAC, and Multiplier at different temperatures, as shown in Figure 9, indicate that despite minor deviations caused by current mirrors, the overall functional form remains intact.

As operating current changes shift circuit activity from the WI to SI regime, measured and simulated power consumption

of an S-AC-based device, shown in Figure 10(a), reveal an increase in power usage. Jennifer Hasler et al. have extensively discussed this concept [3].

Slew rate, bandwidth, input response, and the total available current for charging node capacitance improve as the number of S-AC blocks increases. The reduction in settling time is primarily influenced by hyper-parameter CCC, where a decrease in CCC results in longer settling time, indicating a transition from SI to WI. This occurs because the capacitor at the output transistor's gate requires more time to charge due to limited available current, as explained by A. Nandi et al [4].

The maximum input frequency at which the system can function is determined by its settling time. This is calculated based on the maximum settling time of an S-AC circuit, assuming all operations occur in parallel. According to A. Nandi et al., settling time components such as slew time, dead time, and recovery time must be considered when working current changes cause a transition from WI to SI [4]. As this transition occurs, the time required to charge the capacitance node decreases, allowing the circuit to operate at a higher speed.

Literature Review

In 2023, P. Kumar et al. proposed a bias-scalable CMOS analog processor for machine learning applications in their paper, Bias-Scalable Near-Memory CMOS Analog Processor for Machine Learning [4]. The study highlighted improved scalability and efficiency across various fabrication processes and operating conditions. This work was published in the IEEE Journal on Emerging & Selected Topics in Circuits & Systems, Vol. 13, No. 1. In 2024, Jennifer Hasler et al. demonstrated energy-efficient programmable analog computing using a standard CMOS process in their paper, Energy-Efficient Programmable Analog Compu-ting: Analog Computing in a Standard CMOS Process. Their research showed how such computing could be implemented efficiently with minimal energy consumption. This was published in the IEEE Solid-State Circuits Magazine, Vol. 16, Issue 4.

Also, in 2022, A. Nandi et al. introduced CMOS analog computing circuits that are scalable across variations in process, bias, and temperature in their work, Process, Bias, and Temperature Scalable CMOS Analog Computing Circuits for Machine Learning. Their findings contributed to improving the adaptability of circuits for different environmental and operational conditions. This paper was published in IEEE Transactions on Circuits and Systems I: Regular Papers.

In 2024, Ria Rashid developed a machine learning-driven global optimization framework for analog circuit design, which was presented in her paper, Machine Learning-Driven Global Optimization Framework for Analog Circuit Design. Her research improved efficiency in analog circuit design by leveraging ML techniques. The study was published in Microelectronics Journal, Vol. 151.

Lastly, Y.-H. Wu, W.-H. Lin, and S.-H. Huang presented a low-power hardware implementation for parametric ReLU functions in their 2020 paper, Low-Power Hardware Implementation for Parametric Rectified Linear Unit Function. Their work focused on realizing efficient hardware for deep learning applications, and it was published in the IEEE International Conference on Consumer Electronics (ICCE-Taiwan).

Margin Propagation Algorithm

The confidence measures P_i derived from (1) are applicable to graphs that closely resemble the probability propagation methods outlined in [5]. In this section, we apply it to a specific example of a propagation algorithm, which is commonly referred to as the forward-recursion method in machine learning. We begin by outlining the method in the context of its conventional normalization process. The probability of a state given a fully connected network with C states.

$$i \in \{1,2, \ldots \ldots S\} \qquad (1)$$

at a time, instant n is given by

$$P_i[n] = \sum_j P_j[n-1] P_{ji}[n] \qquad (2)$$

Where $P_i[n]$ is the probability of state i at time n & $P_{ji}[n]$ probability of transition from state j to i at time instant n. The equation computes the probability of state i based on previous state that is $\sum_j P_j[n-1]$ and transaction probability.

Application-specific techniques could be used to produce these transition probabilities, such as neural networks, SVMs, a blend of Gaussians, or table lookups [4,6,7]. The confidence estimations in the margin propagation algorithm may be negative.

Let's consider an intermediate value between i and j at time say n

$$f_{ij}[n]$$

Which are combined through the following recursion:

$$\sum_i [f_{ij}[n] - Z_j[n]] = P_j[n-1] \qquad (3)$$

Eq. (3) combines the intermediate value $f_{ij}[n]$ with the previous Probabilities

$P_j[n-1]$ where $Z_j[n]$ is the normalization constant.
This first step is margin normalization.

$$P_i[n] = \sum_j [f_{ij}[n] - Z_j[n]] \qquad (4)$$

The second step is the accumulation step over the inter-connected graph. Like conventional recursion, the sum of measures is conserved during propagation because

$$\sum_i P_i[n] = \sum_j P_j[n-1], \forall n \qquad (5)$$

So, Equation (5) represents the conservation of all probabilities across various steps of recursion.

A Shape-based Analog Computing (S-AC) paradigm in which the functions that are implemented are resilient to variations in operation temperature and biasing circumstances. S-AC circuits may therefore be run at various speeds and power dissipation levels without altering the output function's characteristics, much like digital circuits. The method for creating S-AC circuits involves first designing a simple proto-function whose "shape" will be independent of MOSFET biasing or operational temperature (within a specified error margin). By developing in-creasingly intricate proto-functions that mirror crucial diode and mosfet actions, we expand on our earlier work in bias-SAC circuits [4]. Other nonlinear and linear approximations can then be obtained by translating, inverting, adding, and subtracting the fundamental proto-function.

Shape-Based Analog Computation

Forward & reverse currents differences could be utilized to determine the drain-to-source current (Ids) passing via an n-type mosfet [8-10].

$$I_{ds} = I_s[f(V_g, V_s) - f(V_g, V_d) \qquad (6)$$

Where f is function and Is is the particular current. The forward and reverse currents are modeled by the function R × R → R in relation to the gate (V_g), drain (V_d), and source (V_s) voltages, respectively.

The following characteristics are always met by the function f(•, •):
• f(0,0)=0 that is zero reference point & f(•, •) is always positive or f(•,•)≥0 by construction.
• f(•, •) is monotonic. For $V_{g1}>V_{g2}$, $f(V_{g1},V_s)>f(V_{g2},V_s)$ & for $V_{s1}>V_{s2}$, $f(V_g,V_{s1})<f(V_g, V_{s2})$.
This shows the monotonic behavior of the function.

SAC aims at designing proto functions that are unbiased & independent by operational circumstances that solely rely on the previously mentioned general features of f(•, •). Here, we outline one technique for producing such a proto-function h: RS → R calculated as solution to equation $h(x)=f(V_B,0)$, here, variable V_B is solution to following equations, given an input vector x ∈ RS with elements $xi \in R$, i=1..., S.

$$\sum_{i=1}^{S} f(V_i, V_B) = C$$

$$f(V_B, 0) - f(V_B, V_i) + f(V_i, V_B) = x_i, \forall i = 1,2, \ldots S \qquad (7)$$

Where C is a parameter that regulate the behavior of proto function h. V_i is the var-iable associated with input x. Despite going into an elaborate mathematical exposi-tion, we could show that h(•) satisfies.

$$1 \geq \frac{\partial h}{\partial x_i} \geq 0, \qquad \forall i$$

The property ensures that the proto-function h is monotonic with respect to its varia-ble

$$\lim_{x_{i \to \infty}} \frac{\partial h}{\partial x_i} = 1$$

identifies the proto-function's two asymptotes regardless of the particular form of f. The nonlinearity may be precisely adjusted by varying hyper-parameter S & vector x, which regulate the transition between the two asymptotes.

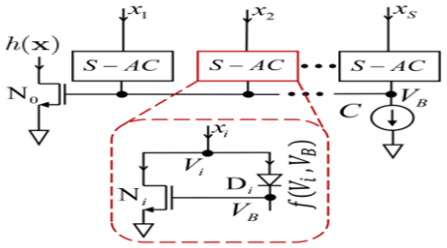

Figure 1: MOS-Based S-AC Implementation [2]

In various operating regimes, the impact of several inputs and hyper-parameter S on proto-function's form could be observed.

S-Ac Circuits for Machine Learning Inference

The essential components for constructing an ML inference processor include non-linear computational circuits, multiply-accumulate (MAC) units, memory for storing inference parameters, chips-in-the-loop training, and digital input interfaces [11].This section demonstrates how these core elements can be realized by modifying or extending the fundamental S-AC circuit depicted in Figure 2. Specifically, a multiply-accumulate (MAC) operation is designed by integrating a non-linear multiplier circuit with compressive mixed-signal memory.

ReLU Implementation Using S-AC

Soft ReLU operations can be derived from one-dimensional proto-functions.

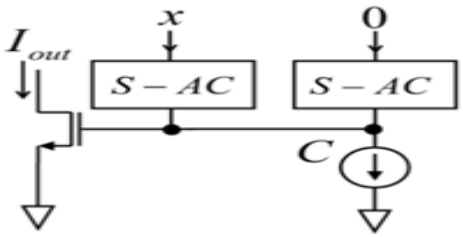

Figure 2: Circuit Implementation of the Soft ReLU Function [2]

As illustrated in Figure 2, the circuit employs two S-AC units for soft ReLU functionality. One unit operates with zero current (or remains floating), while the other processes the input xxx. The geometry of the ReLU function is governed by the constant current CCC. Notably, as CCC approaches zero, the proto-function converges to an ideal ReLU function. The soft ReLU proto-function can be further modified by incorporating output-producing S-AC units. Additionally, fundamental proto-functions, such as the tanh(•) function represented in C1, can be translat-ed, added, subtracted, and shifted to generate a variety of non-linear functions.

Analog Multiplier Based on S-AC

Analog multipliers can be designed using the S-AC proto-function hhh by leveraging Taylor series approximations.

$$h(C+w+C+x)-h(C+w+C-x)+\cdots+h(C-w+C+x)$$
$$-h(C-w+C+x)\sim 2x\,.\left(\frac{dh(C+w)}{dw}-\frac{dh(C-w)}{dw}\right)$$
$$\cong 2x(w^{+}-w^{-})$$

The constant C ensures that the proto-function's input remains positive. Additionally, the differential combination effectively eliminates the second-order and zero-th-order components in the Taylor series expansion [12]. It is important to note that the non-linear mapping dhdw\frac{dh}{dw}dwdh, which exhibits compressive properties, is one of the differential arguments for the multiplier (w+−w−) (w^+ - w^-) (w+−w−). As a result, stored parameters must undergo pre-processing before being used as inputs to the multiplier.

MOS-Based S-AC Multiplier

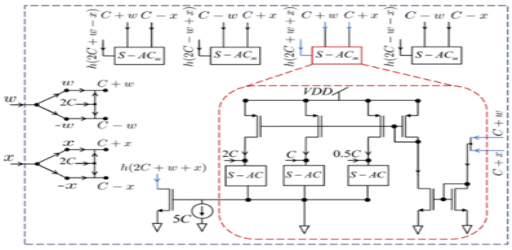

Figure 3: MOS Implementation of S-AC Multiplier [2]

The circuit shown in Figure 3 implements the scalar multiplication described by the equation w∈R, x∈R, y∈Rw \in \ mathbb{R}, x \in \mathbb{R}, y \in \mathbb{R}w∈R, x∈R,y∈R. Each term in the equation is realized through the S-ACm_mm (where the subscript mmm denotes the multiplier) unit. To shift the operation into the first quadrant, a constant CCC is added to the negative term after transforming inputs into their differential forms. The outputs of each S-ACm_mm unit are then combined through addition and subtraction in a differential manner to perform the required multiplication.

Building on this foundational approach, inner-product and multiply-accumulate (MAC) operations can be realized by integrating multiple S-AC multipliers with summing circuits, leveraging Kirchhoff's current law for computation.

Compressive Memory with S-AC

A key challenge in implementing analog machine-learning processors is the efficient storage and updating of trained parameters. While nanoscale device-based analog memory—such as memristors and floating-gate transistors—has been proposed for this purpose, these solutions face limitations in scalability concerning inference and training speeds.

To address this, a compressive function can be developed using a DAC-based memory with an analog front-end based on S-AC. This design aligns with the multiplier's requirements, ensuring effective parameter storage and retrieval while maintaining the efficiency needed for machine-learning applications.

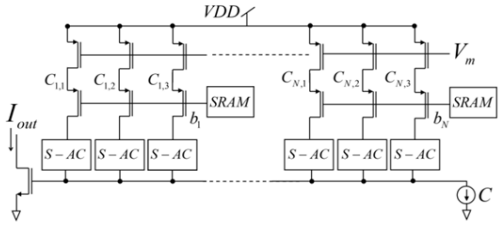

Figure 4: SAC Memory Implementation

S-AC Based Compressive Memory DAC

The proposed DAC is built on an 8-bit S-AC structure, designed to handle decimal inputs ranging from 0 to 255 across various operating conditions. Throughout different operational regimes, both the output characteristics and the ideal response remain consistent [12].

The functionality of the S-AC-based DAC transitions from Weak-Inversion (WI) to Strong-Inversion (SI) when S=3S = 3S=3. This shift enhances speed and throughput by reducing the settling time; however, it also increases power consumption. Despite this, the Moderate-Inversion (MI) operating zone provides the best balance between energy efficiency and throughput, making it the optimal trade-off for performance.

Result

The goal is to multiply two voltages, v1v_1v1 and v2v_2v2. The voltage v1v_1v1 operates at a frequency of 100 MHz and has a peak-to-peak voltage of 4 Vpp, while v2v_2v2 has a frequency of 10 MHz and a peak-to-peak voltage of 2 Vpp. The multiplier circuit produces an output signal with a peak-to-peak amplitude of 5.6 V, which is consistent with the theoretical calculation. The S-AC multiplier circuit was tested under various conditions and for different values of the design parameter SSS, as illustrated in Figure 5. As the value of SSS increases, the accuracy of the multiplier improves, approaching ideal performance.

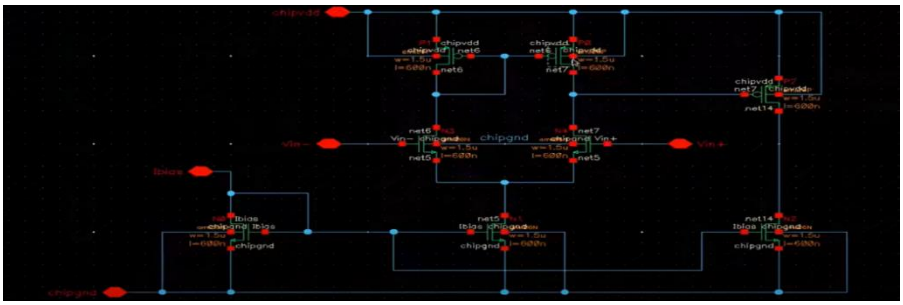

Figure 5: Implementation of S-AC MOS

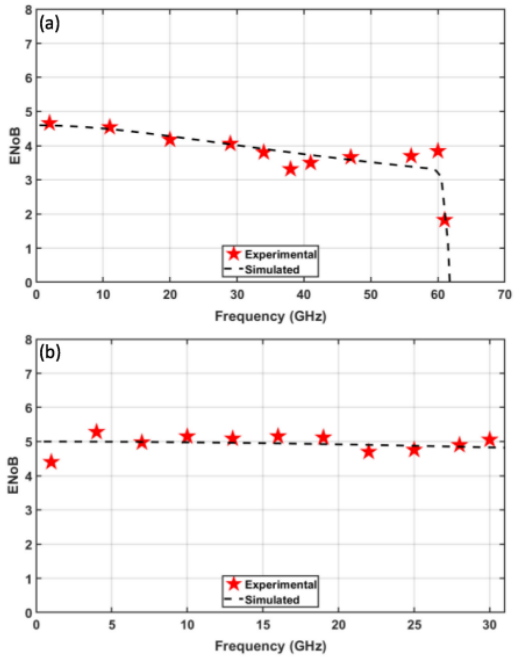

Figure 6: Effective No.of Bits

By modifying the design parameter SSS, the S-AC design enables a range of trade-offs between power, area, and accuracy. The design and form analysis are illustrated in Figure 10. The value of SSS must be carefully chosen based on the application's specific requirements. Theoretically, selecting splines within a range of 1 to SSS offers several trade-offs. Analysis showed that each increase in the design parameter SSS results in an exponential decrease in approximation error, but this comes at the cost of approximately 22% more area compared to the previous value of SSS. Performance evaluations of the S-AC multiplier indicated that setting S=3S = 3S=3 can lead to up to 33% area savings and 35% power reduction, while maintaining an average absolute error of just 4%.

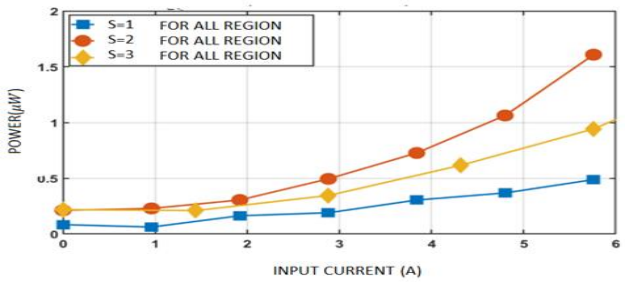

Figure 7: Design Margin and Shape Analysis

Design Margin and Shape Analysis

The analysis reveals that a design parameter of S=4S = 4S=4 ensures that the desired shape is maintained with over 98% accuracy. In contrast, configurations with S=1S = 1S=1 and S=3S = 3S=3 still achieve high classification accuracy but may exhibit slight variations. Importantly, the system retains its accuracy as the network learns, even under hardware con-straints. The shape-based analog computing (S-AC) framework is specifically designed to maintain its transfer function within a strict error margin under varying biasing conditions—Weak-Inversion (WI), Moderate-Inversion (MI), and Strong-Inversion (SI)—as well as under temperature fluctuations. This capability ensures that the

system accurately replicates the desired functional form in machine learning applications.

S-AC designs prioritize selecting design parameters based on application requirements, shifting the focus from conventional design techniques to optimizing the desired functional shape. This flexibility allows users to adjust S, balancing accuracy, energy efficiency, and area constraints, similar to digital design methodologies [13].

Energy and Error Analysis

Unwanted effects, such as mismatches, noise, and power-supply variations, can introduce additive errors that impact circuit performance. In S-AC circuits, the margin between the shapes obtained in SI and WI regimes accounts for variations caused by second-order effects.

Operation	VDD = 1V	EnergyConsumption (pJ)
	WI	MI
ReLU	0.49	2.01
Multiplier	0.78	3.02
S-AC	0.23	1.02

Table 1: Energy Consumption Per Operation Unit

From the table, it is evident that as the operating regime shifts from WI to SI, energy consumption per operation increases. However, the MI region provides an optimal trade-off between power efficiency and computational speed. The most significant errors in S-AC circuit operations stem from mismatches, noise, and power-supply variations. Despite these challenges, S-AC circuits effectively preserve the inherent functional shape of the implemented computation.

Performance Analysis Temperature Scalability

To assess temperature scalability, we analyze the impact of nominal temperature variations on different S-AC units. Figure presents measured characteristic curves for S-AC-based ReLU, Multiplier, and DAC across varying temperature conditions.

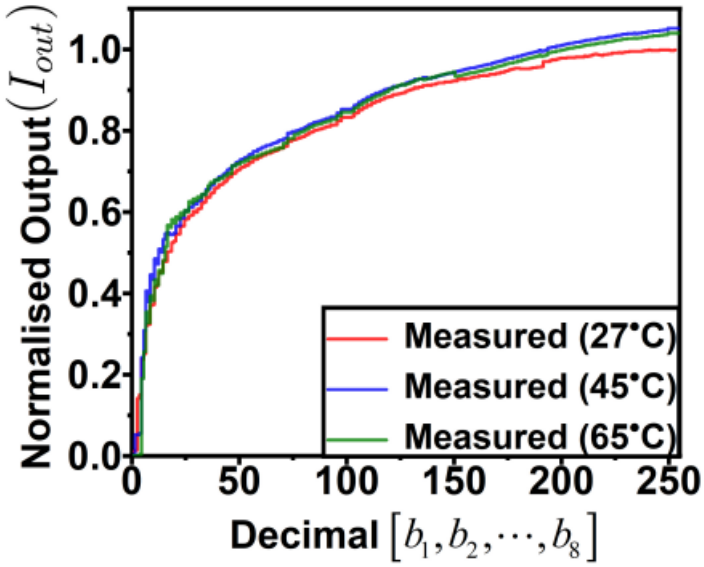

Figure 8: Temperature Variation of SAC

Despite minor variations attributed to current mirror effects, the overall functional shape remains intact, reinforcing the robustness of the S-AC design. The most significant operational errors in S-AC circuits arise from mismatches, noise, and power-supply variations. These errors can introduce additive effects, potentially degrading circuit performance. The margin between WI and SI regimes accounts for all second-order variations, preserving the intended shape-function.

Conclusion

The S-AC-based framework consistently maintains its transfer function within a narrow error margin, even under varying biasing conditions (WI, MI, and SI) and temperature fluctuations. This paper includes a Monte Carlo analysis to demonstrate that gate-source voltages and drain-source currents do not affect the performance of the MOS pair. The proto-shape is controlled by the design parameter SSS, which is crucial for machine learning applications, as it determines the system's ability to accurately replicate the desired functional form. As noted by Shrivastava et al. and Joshi et al., allowing users to select the proto-shape based on the value of SSS and focusing on achieving the intended functional forms instead of relying on traditional design methods offers greater flexibility [14,15]. This approach further reduces the precision constraints typically as-sociated with computing. Additionally, by adjusting SSS, users can balance computational accuracy with energy consumption and area, similar to the trade-offs in digital designs, as shown by

Huyang et al. [16-18].

References

1. Kumar, P., Nandi, A., Chakrabartty, S., & Thakur, C. S. (2023). Bias-scalable near-memory CMOS analog processor for machine learning. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 13(1), 312-322.
2. S. Chakrabarty, "Margin decoding communication system," U.S. Patent 8 060 810, Nov. 15, 2011.
3. Hasler, J. (2024). Energy-Efficient Programable Analog Com-puting: Analog computing in a standard CMOS process. IEEE Solid-State Circuits Magazine, 16(4), 32-40.
4. Kumar, P., Nandi, A., Chakrabartty, S., & Thakur, C. S. (2022). Process, bias, and temperature scalable CMOS analog computing circuits for machine learning. IEEE Transactions on Circuits and Systems I: Regular Papers, 70(1), 128-141.
5. Xiao, T. P., Bennett, C. H., Feinberg, B., Agarwal, S., & Mari-nella, M. J. (2020). Analog architectures for neural network acceleration based on non-volatile memory. Applied Physics Reviews, 7(3).
6. Wu, Y. H., Lin, W. H., & Huang, S. H. (2020, September). Low-power hardware implementation for parametric rectified linear unit function. In 2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan) (pp. 1-2). Ieee.
7. Kumar, P., Zhu, K., Gao, X., Wang, S. D., Lanza, M., & Thakur, C. S. (2022). Hybrid architecture based on two-dimensional memristor crossbar array and CMOS integrated circuit for edge computing. npj 2D Materials and Applica-tions, 6(1), 8.
8. Rashid, R., Krishna, K., George, C. P., & Nambath, N. (2024). Machine learning driven global optimisation framework for analog circuit design. Microelectronics Journal, 151, 106362.
9. Demler, M. (2018). Mythic multiplies in a flash. Microprocesser Report.
10. Thakur, C. S., Wang, R., Hamilton, T. J., Tapson, J., & van Schaik, A. (2016). A low power trainable neuromorphic inte-grated circuit that is tolerant to device mismatch. IEEE Transactions on Circuits and Systems I: Regular Pa-pers, 63(2), 211-221.
11. Freund, K. (2021). IBM Research Says Analog AI Will Be 100X More Efficient. Yes, 100X. Yes 100X, Sep.
12. Srivastava, P., Yadav, R., & Srivastava, R. (2020). Ultra-high speed and novel design of power-aware CNFET based MCML 3-bit parity checker. Analog Integrated Circuits and Signal Processing, 104, 321-329.
13. Joshi, V., Le Gallo, M., Haefeli, S., Boybat, I., Nandakumar, S. R., Piveteau, C., ... & Eleftheriou, E. (2020). Accurate deep neural network inference using computational phase-change memory. Nature communications, 11(1), 2473.
14. Huang, G., Hu, J., He, Y., Liu, J., Ma, M., Shen, Z., ... & Wang, Y. (2021). Machine learning for electronic design automation: A survey. ACM Transactions on Design Automation of Electronic Systems (TODAES), 26(5), 1-46.
15. Chandrasekaran, S. T., Jayaraj, A., Karnam, V. E. G., Banerjee, I., & Sanyal, A. (2021). Fully integrated analog machine learning classifier using custom activation function for low resolution image classification. IEEE Transactions on Cir-cuits and Systems I: Regular Papers, 68(3), 1023-1033.
16. Praveen, J., Khawaja Nizammuddin Subhani, Dsouza Deepakraj Peter, Kanthesh Jogi, and D. N. Darshan. "Design and implementation of low power dynamic buffer circuit for nanotechnology application." Materials Today: Proceedings 35 (2021): 461-464.
17. Srivastava, P., Yadav, R., & Srivastava, R. (2022). Low-Power and High-Speed Design of FinFET-Based MCML Delay Element. In Proceedings of First International Conference on Computational Electronics for Wireless Communications: ICCWC 2021 (pp. 185-194). Springer Singapore.
18. Nair, A. R., Nath, P. K., Chakrabartty, S., & Thakur, C. S. (2022). Multiplierless MP-kernel machine for energy-efficient edge devices. IEEE Transactions on Very Large-Scale Integra-tion (VLSI) Systems, 30(11), 1601-1614.