Data-Driven Performance Evaluation of Analog Computing Circuits: Shape and Margin Analysis with Machine Learning

Abhishek Agwekar[1], 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*

Abstract. This paper presents a novel approach to the design and analysis of shape-based analog computing (S-AC) circuits, leveraging machine learning techniques to enhance performance in VLSI technologies. The study explores the key features of S-AC circuits, particularly their scalability, precision, speed, and power efficiency, as compared to conventional digital designs. Machine learning architectures are integrated with mathematical function approximations to optimize the implementation of S-AC circuits, enabling efficient circuit simulations and performance evaluations. The input-output characteristics are mapped using a CMOS process, ensuring adaptability across different process variations. Notably, S-AC-based neural networks exhibit high resilience to temperature fluctuations, maintaining consistent accuracy. Furthermore, the study introduces a design margin and shape analysis framework, where the design parameter (S) and machine learning models play a crucial role in ensuring functional accuracy. Unlike traditional circuit design methods, the S-AC approach allows users to select prototype shapes based on specific application requirements, optimizing functional forms dynamically through machine learning. This research highlights the potential of machine learning-driven analog circuit design to advance scalable and efficient VLSI implementations, paving the way for next-generation computing architectures.

Keywords: Process Scalability, Margin Propagation, Machine Learning, S-AC Computing, VLSI, Machine Learning.

1. INTRODUCTION

In the MI regime, power and speed will always be balanced optimally. Devaluation of Performance Mismatches, noise, and power-supply fluctuations are the three main faults that are introduced into the functioning of S-AC circuits. Additive mistakes can significantly impair the circuits' functioning as a result of these undesirable impacts. The gap between the forms seen in SI & WI regimes in S-AC circuits accounts for all of the differences brought on by second-order effects. This important characteristic enables the S-AC circuits to maintain the intrinsic form of the function that is being implemented[1],[2].

1.1 Temperature Variation

Here study to examine how S-AC units are affected by nominal temperature variations. S-AC based ReLU,
DAC, & Multiplier observed characteristic curves at various temperature points are displayed in Fig. 9. It is evident that the general distinctive form is maintained despite a minor change in the intended curves that can be ascribed to the existing mirrors.

1.2 Power & Task-Energy Efficiency

As shown in Fig. 10a compares the measured along with simulated power of an S-AC-based device when operating current is altered to shift circuit activities from WI to SI. Power consumption increases as circuit operation transitions from WI to SI.

1.3 Slew Rate

As the S-AC block number improves slew rate, bandwidth, inputs, & total current available to charge node capacitance. Only the limitations imposed by the hyper-parameter C can be responsible for this overall decrease in settling time. The settling time increases as the hyper-parameter C decreases, indicating a transition from the Strong Inversion (SI) to the Weak Inversion (WI) regime,

as the capacitor at the output transistor's gate requires more time to charge due to the restricted available current.

1.4 Settling Time

This settling time, which is determined by calculating max settling time of an S-AC, determines the maximum input frequency at which the system may function (assuming all actions are conducted in parallel). Dead time, slew time, as well as recovery time-based units, have been included when the working current is altered to cause the circuit to transition from the WI to the SI area of operation. As the operating regime changes from WI to SI, it is clear that charging the capacitance node takes less time. As a result, the circuit can run faster.

2. MARGIN PROPAGATION ALGORITHM

The confidence measures P_i derived from (1) are applicable to graphs that closely resemble the probability propagation methods outlined in [8]. 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.

$$i \in \{1,2, \dots \dots . S\} \quad (1)$$

at a time, instant n is given by

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

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 [9], SVMs [10, 4], a blend of Gaussians, or table lookups. 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:

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

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.

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

The second step is the accumulation step over the interconnected graph.

Like conventional recursion, the sum of measures is conserved during propagation because

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

So, Eq. (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 increasingly 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.

2 Shape-Based Analog Computation

Forward & reverse currents [8] differences could be utilized to determine the drain-to-source current (I_{ds}) passing via an n-type MOSFET.

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

Where f is function and I_s is the particular current. The forward and reverse currents are modeled by the function $R \times R \rightarrow R$ 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 $V_{s1}>V_{s2}$, $f(V_g,Vs1)<f(V_g, V_{s2})$.

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.

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

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

Where C is a parameter that regulate the behavior of proto function *h*. V_i is the variable associated with input *x* . Despite going into an elaborate mathematical exposition, 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 variable

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

$$\lim_{x_i \to -\infty} \frac{\partial h}{\partial x_i} = 0$$

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.

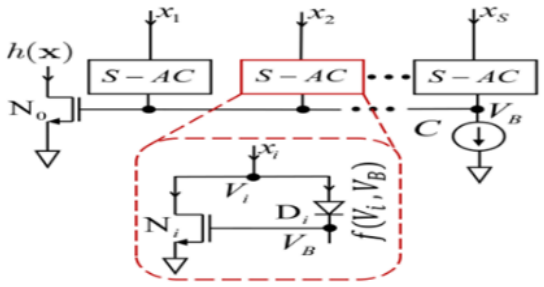

Fig. 1. MOS-based S-AC implementation [2]

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

3 S-AC CIRCUITS FOR MACHINE LEARNING INFERENCE

Fundamentals for building an ML inference processor are non-linear computing circuits, multiply-accumulate circuits and memory storing inference parameters, chips-in-the-loop training, & digital input interfaces. [5]

Here, demonstrate how all of the building blocks may be implemented by modifying or expanding the simple S-AC circuit seen in Fig 2. In particular, we develop a multiply-accumulate (MAC) operation by combining a non-linear multiplier circuit with compressive mixed-signal memory.

3.1 ReLU Implementation with S-AC

Soft ReLU operations could be developed via one-dimensional proto-functions.

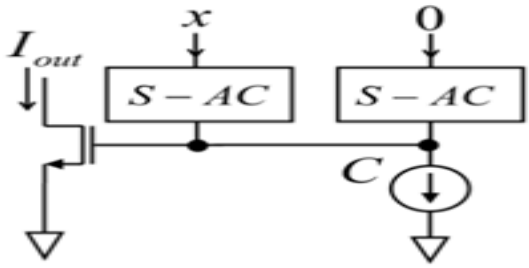

Fig. 2. Implementation of the soft ReLU [2]

Fig.2 displays a circuit implementation of the soft ReLU function. Two S-AC units are used in the basic circuit; one is controlled by zero current (or floating), while the other receives an input x. The geometry of the ReLU is determined by the constant current C. It should be noted that the proto-function converges to an ideal ReLU function at the limit $C \to 0$.

By integrating output-producing S-AC units, the proto-function with soft ReLU could be modified. Additionally, keep in mind that the fundamental proto-function, such as the tanh(·) function shown in C_1, may be translated, added, subtracted, and shifted to create additional non-linear functions. An analog multiplier based on S-AC The following Taylor series approximations can be used to create analog multipliers using the S-AC proto-function *h*.

$$h(C+w+C+x)-h(C+w+C-x)+\cdots+h(C-w+C+x)-h(C-w+C+x)\sim 2x\,.\left(\frac{dh(C+w)}{dw}-\frac{dh(C-w)}{dw}\right)\cong 2x(w^{+}-w^{-}) \quad (8)$$

where C is the Hyper Parameter controlling non linearity

w is the input variable

x is another input variable.

The proto-function's input is always positive thanks to the constant C. The second-order and zero-th-order components in the Taylor series are essentially eliminated by the differential combination [6]. Be aware that non-linear map dh dw, compressive map according to its properties, is among the differential arguments to multiplier (w+−w−). Therefore, before being used as an input to the multiplier, the stored parameters must be pre-processed.

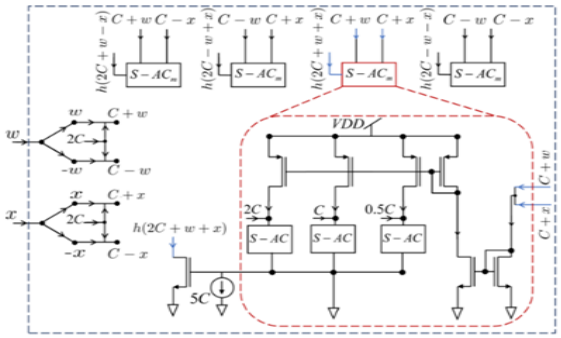

Fig. 3. MOS implementation of S-AC multiplier [2]

The scalar multiplication provided in the equation w ∈ R, x ∈ R, and product y ∈ R is implemented by the circuit in Fig. Each component of the equation is implemented using the S-AC$_m$ (subscript m for multiplier) unit shown in Fig. 3 Constant(C) is added to negative term after inputs have been converted into their differential forms in order to transfer operation into first quadrant. Output from each S-AC$_m$ unit is added and subtracted (differentially) in line to accomplish the required multiplication.

On basis of this basic procedure, inner-products & multiply-accumulate operations may be constructed by combining element-wise S-AC multipliers with summing circuits based on Kirchhoff's current law.

3.2 Compressive Memory with S-AC

The storing and updating of trained parameters are a significant implementation difficulty for analog machine-learning processors. Although memristor-, floating-gate-, and other nanoscale device-based analog memory have been suggested for analog machine-learning processors, their speed and functional responsiveness do not scale over inference and training. To create a compressive function using a DAC-based memory that employs an analog frontend based on S-AC, as needed by the multiplier in the equation.

$$g(x)=\log_2\left(\sum_i 2^{x_i}\right) \quad (9)$$

Verifying that g(x) meets the conditions is therefore simple.

$$1\geq\left|\frac{\partial g}{\partial x_i}\right|\geq 0$$

$$\lim_{x_i\to\infty}\frac{\partial g}{\partial x_i}=1$$

The proto-function *h (x)* is comparable with *g(x)*

If the binary representation of x is as follows:

$$x\cong\sum_{i=1}^{N}2^{i}b_i \quad (10)$$

Where b_i can be {0,1} and are binary bits of *n*

The hyper-parameter S is then increased by one bit, giving us.

$$g(x)=\log_2\sum_{i=1}^{N}\sum_{j-1}^{S}2^{C_{ij}}b_i=\log_2\sum_{i,j:b_i=1}^{N\,S}2^{C_{ij}}=g(B)$$

Where C_{ij} is the offset value of each bit.

$$B\ \in\ \{0,1\}^{S}\,X\,\{0,1\}^{N}$$

N denotes number of inputs, and N is a binary input matrix. *S* is the no. of offset input bits. One can use the proto function *h (x)* to approximate it.

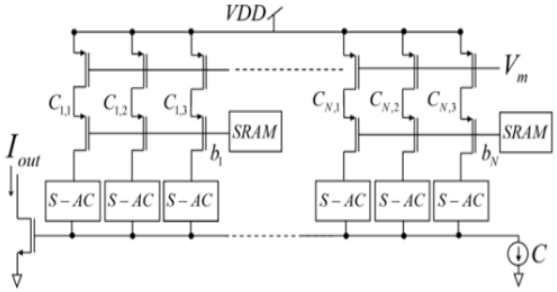

Fig. 4. Implementation using S-AC memory [10]

The N-bit S-AC based compressive memory circuit implementation for S=3 is depicted in Fig. Transmission gate (TG) switches are used to build switches connected at $b_1, b_2, \ldots, b_N$. In this case, each N-bit binary number to

be translated into its analog counterpart is represented by [b_1,...., b_{N-1}, b_N], and the offsets for S=3 are [$C_{1,1}$,$C_{1,2}$,$C_{1,3}$,...,$C_{N,3}$]. The digital input is transformed into a compressed analog output by the suggested S-AC-based DAC. It should be noted that the equation for multiplication inherently expands this compressive result.

3.3 S-AC Based Compressive Memory DAC

DAC is based on 8-bit S-AC with function of comparable decimal input that ranges 0-255 under various operating conditions. Both the output shape and the ideal form are constant throughout operation regimes[6] .S-AC-based DAC operation shifts from WI to SI for S=3, increasing power consumption while decreasing settling time, which enhances throughput and speed. Nonetheless, the MI zone of operation offers optimal trade-off within energy & throughput.

4 RESULT

For two voltages, v_1 and v_2, to be multiplied. The frequency is 100 MHz and the voltage is 4 V_{pp}. The frequency of v_2 is 2 V_{pp} 10 MHz. The circuit multiplies the v_1 and v_2 voltages to produce the output wave. 5.6 V_{pp} is the output amplitude. It is evident that the result matches the theoretical outcome.

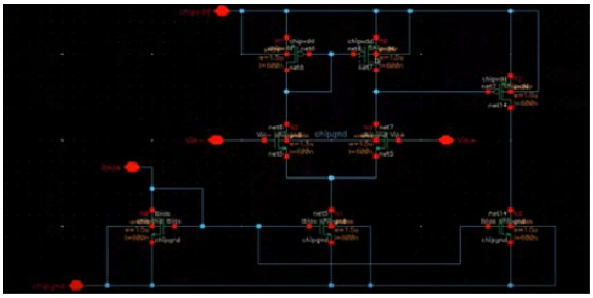

Fig. 5. Implementation of S-AC MOS

The S-AC multiplier circuit was built under various operating circumstances and for varying values of the design parameter S. The multiplier accuracy rises and approaches the ideal as design parameter S grows.

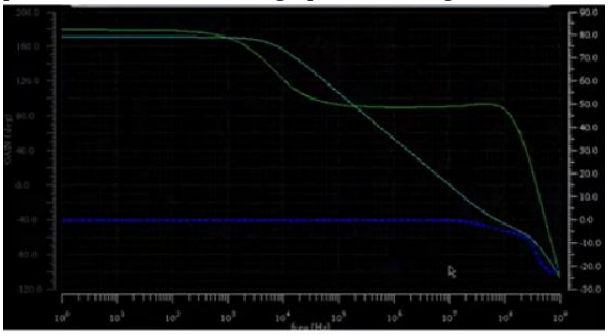

Fig. 6. Timing graph of S=1, S=2, S=3

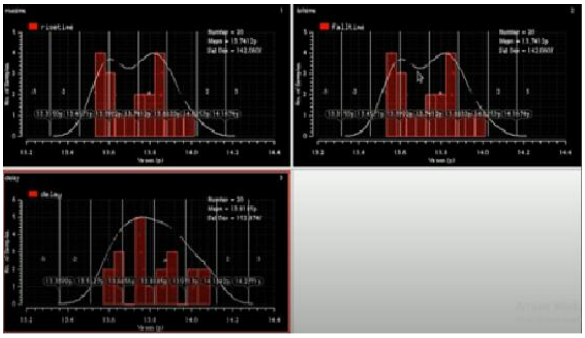

Fig. 7. Delay graph with no. of bits (Samples)

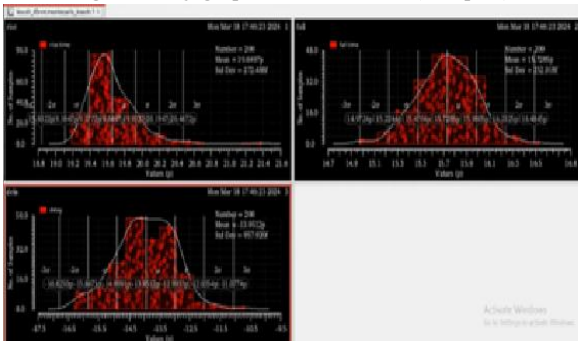

Fig. 8. Monte-Carlo Graph with different values

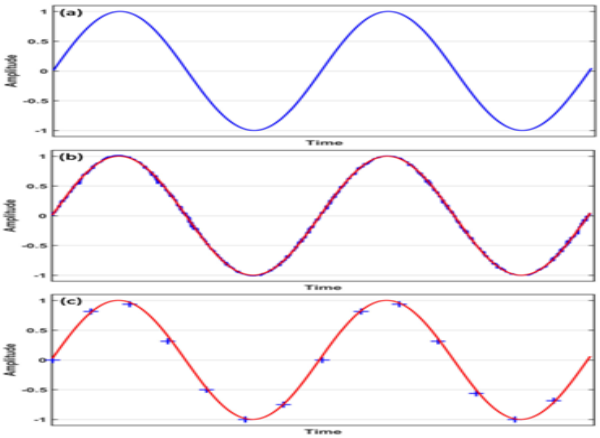

Fig. 9. Signal progression through the ADC model

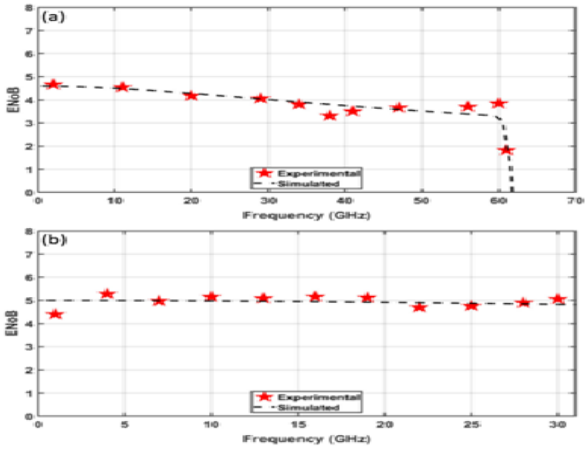

Fig. 10. Effective no. of bits

S-AC design offers a range of trade-offs between accuracy, area, and power improvements by altering the design parameter (S). The value of this design parameter, S, is determined by the requirements of the application. There is theoretically a large variety of trade-offs to pick from because the number of splines chosen can range from 1 to S. Analysis revealed that every increase in design parameter S results in an exponential reduction in approximation error at the expense of an extra ≈ 22% area need as compared to prior S. According to the S-AC multiplier's performance evaluation, S = 3 may save up to 33% of the area and up to 35% of the power, with an average absolute inaccuracy of 4%.

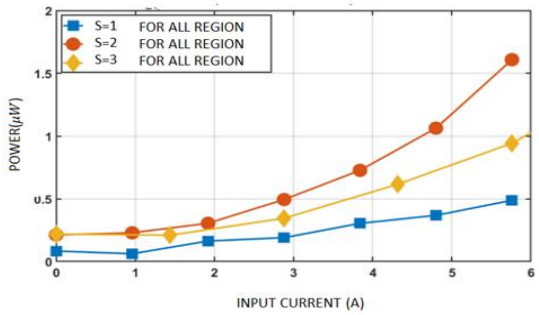

Fig. 11. Design Margin and Shape Analysis

Energy Consumption Analysis

An result of these undesired effects, the functionality of the circuits can be severely affected by additive errors. In S-AC circuits, the margin between the shapes obtained in the SI and WI regimes takes into account all the variations due to second-order effects.

Table –I Energy per operation unit

Operation VDD=1V	Energy Consumption		
	WI	MI	SI
ReLU	0.49	2.01	3.00
Mutiplier	0.78	3.02	5.05
S-AC	0.23	1.02	2.45

Table I shows the energy per operation of basic operations mapped in the shape-based analog computation.

It can be seen that as the circuit operating regimes move from WI to SI, the energy per operation increases while an optimal balance between power and speed is always obtained in MI region.

The most significant errors introduced in the operation of S-AC circuits are represented by mismatches, noise and power-supply variations.

It can be observed that the

- The non-linear shape remains invariant in weak, moderate and strong inversion regimes as desired.
- Normalized output current curve follows the desired non-linear shape and matches with the ideal.

Temperature Variation

We compare the effect of nominal temperature variation on S-AC units. One can observe that even though there is a slight variation that can be attributed to the current mirrors in the desired curves, but the overall characteristic shape is preserved.

Power Consumption

- A a comparison plot between measured and simulated power of S-AC based unit when the operating current is varied such that circuit operations moves from WI to SI regime.
- The power consumption increases when circuit operation shifts from WI to SI regime.

5 CONCLUSION

Under various biasing circumstances (WI, MI, and SI) and temperature fluctuations, the shape-based analog computing framework is made to maintain its transfer function within a strict error margin. The monte Carlo analysis is done in this paper. The pair of MOS is unaffected by the drain-source currents or the gate-source voltages. The design parameter S determines the system's capacity to precisely duplicate the intended functional form and controls this proto-shape, which is significant in machine learning applications.[7].

By letting user select proto-shape (by selecting design parameter S) on the basis of application's requirements and concentrating on achieving desired functional forms rather than traditional design methodologies, S-AC design further loosens the constraints of precision computing. Additionally, by adjusting S, user may trade off computational accuracy with energy and area, much like with digital designs [11].

REFERENCE

[1] P. Kumar et. al., “Bias scalable near memory CMOS Analog processor for Machine learning” IEEE Journal on emerging and selected topics in circuits and systems Vol 13, No. 1 March 2023.

[2] S. Chakrabarty et. al., "Margin decoding communication system," U.S. Patent 8 060 810, Nov. 15, 2011.
[3] Jennifer Hasler et. al., "Energy-Efficient Programable Analog Computing: Analog computing in a standard CMOS process", IEEE Solid State Circuit Magazine vol 16, issue4, 2024
[4] A. Nandi, et. al., "Process, bias, and temperature scalable CMOS analog computing circuits for machine learning," IEEE Trans. Circuits Syst. I, Reg. Papers, early access, Nov. 1, 2022, Doi: 10.1109/TCSI.2022.3216287.
[5] Ria Rashid et. al., "Machine learning driven global optimization framework for analog circuits design" Microelectronics journal Vol 151 ,2024
[6] M. Delmer, et. al."Mythic multiplies in a flash," Linley Group Microprocessor Rep., Aug. 2018. [Online]. Available: https://www.linleygroup.com/mpr/article.php?id=12023
[7] C. S. Thakur et. al., "A low power trainable neuromorphic integrated circuit that is tolerant to device mismatch," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 63, no. 2, pp. 211–221, Feb. 2016.
[8] T. P. Xiao et. al., "Analog architectures for neural network acceleration based on non-volatile memory," Appl. Phys. Rev., vol. 7, no. 3, Sep. 2020, Art. no. 03130
[9] Y.-H. Wu et. al., "Low-power hardware implementation for parametric rectified linear unit function," in Proc. IEEE Int. Conf. Consum. Electron. (ICCE-Taiwan), Sep. 2020, pp. 1–2.
[10] P. Kumar et.al., "Hybrid architecture based on two-dimensional memristor crossbar array and CMOS integrated circuit for edge Computing", 2D Material Applications FCT Nova, 2018
[11] K. Freund. (Sep. 23, 2021). IBM Research Says Analog AI Will be 100X More Efficient.Yes,100X.[Online].https://www.forbes.com/sites/karlfreund/2021/09/23/ibm-research-saysanalog- ai-will-be-100x-more-efficient-yes-100x/?sh=61b5e23b129b