IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 August 2024, accepted 11 August 2024, date of publication 14 August 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3443343

RESEARCH ARTICLE

Yield Maximization of Flip-Flop Circuits Based on Deep Neural Network and Polyhedral Estimation of Nonlinear Constraints

SAYED ALIREZA SAJJADI[1], (Member, IEEE), SAYED ALIREZA SADROSSADAT[2], (Senior Member, IEEE), ALI MOFTAKHARZADEH[1], MORTEZA NABAVI[3], (Member, IEEE), AND MOHAMAD SAWAN[4], (Life Fellow, IEEE)

[1]Department of Electrical Engineering, Yazd University, Yazd 8915818411, Iran
[2]Department of Computer Engineering, Yazd University, Yazd 8915818411, Iran
[3]Department of Electrical and Computer Engineering, Polytechnique Montréal, Montreal, QC H3A 0E9, Canada
[4]Center of Excellence in Biomedical Research on Advances-on-Chips Neurotechnologies (CenBRAIN Neurotech), School of Engineering, Westlake University, Hangzhou, Zhejiang 310024, China

Corresponding author: Ali Moftakharzadeh (moftakharzadeh@yazd.ac.ir)

ABSTRACT In this paper, we propose a method based on deep neural networks for the statistical design of flip-flops, taking into account nonlinear performance constraints. Flip-flop design and manufacturing are influenced by random variations in the technological process, making deterministic design approaches inadequate for achieving high yields. The conventional yield maximization method using Monte Carlo (MC) simulation is a time-consuming process. Also, for many performance constraints, either there are no analytical formulations or if they exist, they are not sufficiently accurate to be used in circuit optimization. To address these challenges, we approximated the nonlinear constraints with linearized ones (polyhedral approximation) and performed a yield maximization process which was done by developing our first proposed method. Then in the second proposed method, we used deep neural networks to generate precise nonlinear closed-form models for circuit performance metrics and also replaced MC simulation with an analytical yield formula. The combination of these techniques significantly enhances the speed and accuracy of statistical circuit design by employing powerful gradient-based optimization methods that converge quickly to the optimal solution. Experimental results demonstrate that our proposed approach enables the design of circuits with various performance constraints under process variation, and achieves more optimum results with much fewer iterations and less CPU time compared to the conventional simulation-based yield maximization methods.

INDEX TERMS Computer-aided design (CAD), circuit yield maximization, circuit simulation, deep neural network (DNN), flip-flop circuits, gate sizing, nanometer regime technologies, process variations, statistical design.

I. INTRODUCTION

While advanced technology nodes offer faster and more complex Systems-on-Chips (SoCs), their manufacturing process becomes more intricate due to more random parameters, leading to a reduction in yield and uncertain behavior of the SoCs [1]. The reliability of SoC design is typically evaluated using a yield metric, which reflects the quality of the manufacturing process [1], [2], [3]. Yield losses can be classified into catastrophic and parametric ones [1]. The catastrophic yield loss occurs when a malfunction causes a complete shutdown of a part of an integrated circuits (ICs) [1]. On the other hand, the parametric yield loss refers to cases where the SoC performance is correct but fails to meet specific metrics such as power consumption or operating frequency [1]. This paper focuses on addressing the parametric yield loss.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Hossein Moaiyeri.

The Monte Carlo (MC) simulation is one of the commonly employed methods to estimate circuit yield under process voltage temperature (PVT) variations before the manufacturing process [2], [3], [4]. The accuracy of MC simulation is directly influenced by the number of simulations conducted [2], [3], [4]. Consequently, many time-consuming MC simulations are needed to predict the manufacturing yields of SoCs [2], [3], [4]. Due to the significance of increasing yield and design optimization speed and accuracy in the chip manufacturing industry, researchers have shown great interest in developing faster yield estimation methods [4], [5], [6] and new yield optimization approaches [2], [7], [8].

Flip-flops are essential components of SoCs as they control the data flow. They play a crucial role in various blocks within an SoC, including frequency dividers and clock networks [9], [10], [11]. By effectively managing data flow, flip-flops contribute to the proper functioning of the SoC. So, any violation of timing and power constraints in flip-flops can result in incorrect SoC functionality and yield losses [11]. Additionally, flip-flops consume a substantial portion (around 40%) of the energy, power, and timing resources in SoCs, highlighting the importance of maximizing their yield [9], [12]. This paper validates the proposed yield maximization process using the partially static high-frequency eighteen-transistor hybrid topological flip-flop (PHFF) [12] and the low-voltage and low-power true-signal-phase sixteen-transistor flip-flop (LLTFF) [9]. The 16 nm predictive technology model (PTM) complementary metal-oxide semiconductor (CMOS) [13] is used to align the fabrication technology with today's technological requirements.

A two-step approach for gate sizing based on aging and process variation was presented in [1]. This method optimizes circuit timing yield and decreases delay degradation due to PVT variation. This method employs approximate mathematical formulas to generate performance metrics that influence the final accuracy. In [14], an advanced Lagrangian relaxation optimization integrated with other approaches like flip-flop sizing and gate merge transformation was used. Although it relaxes certain constraints to simplify the problem, this can lead to suboptimal solutions. The Multi-Armed Bandit (MAB) model was employed in each Lagrangian relaxation epoch to select a suitable optimization heuristic from the interleaved optimization methods. Statistical gate sizing techniques were developed in [3] to approximate delay degradation by considering process variation, followed by incremental gate sizing to identify the optimal gates for the optimization procedure. However, the first-order delay model can impact the accuracy of the approach. In [15], a method for recognizing delay violation in carbon nanotube field-effect transistors (CNFETs)-based circuits under process variation is introduced. In [4], a new approach for yield prediction based on nonparametric statistics was presented. Instead of using complete MC simulations, this method utilized kernel density estimation. An analytical-based method was proposed in [5] to estimate the failure rate of SRAM hold stability in subthreshold regimes under process variation which used the simplified form of transistor equations. In [6], an approach for yield estimation in Analog/Mixed-Signal (AMS) circuits was proposed, employing Joint Recurrence Verification (JRV) on important circuit parameters that significantly impact yield value. A Piecewise Distribution Model (PDM) was proposed in [16] to model the behavior of AMS circuits in high sigma distribution (around 4.8σ). However, achieving acceptable accuracy in yield estimation, particularly at low voltage levels, is challenging. A mathematical formula for circuit delay considering process variation in the subthreshold domain was presented in [17]. One potential limitation of the presented methodology is that it neglects the effect of input-to-output coupling capacitance on the inverter delay, which can restrict the accuracy of yield estimation. The authors in [18] introduced an analytical formula based on the over-capacitance ratio of NMOS and PMOS transistors. This formula enhances circuit sizing speed by considering the Parallel Transistor Stack in low-voltage and subthreshold regions. In [19], a statistical circuit analyzer based on modified polynomial chaos (PC) expansion was introduced which is capable of analyzing circuit variation with non-Gaussian and Gaussian distributions. The state space formula was used in [20] to consider the Read Static Noise Margin (SNM) of SRAMs as a function affected by the threshold voltage. This equation indicates the dependency of the Read SNM on PVT variations, which can help to increase the accuracy of circuit reliability analysis. However, this method is limited to the Read SNM of 6T SRAM cells. An absolute shrinkage deep kernel learning method was proposed in [21] to approximate circuit yield faster than MC simulation by using kernel as a surrogate model (SM), applicable to high-dimensional problems. This method can be applied to high-dimensional problems as well. In [22], a maximum likelihood estimate (MLE) was suggested as a faster alternative to MC simulation in the circuit simulator.

In [2], a yield enhancement algorithm based on freeze-thaw Bayesian optimization (BO) was developed which was shown to be twice as fast as state-of-the-art methods [23]. Authors in [7] utilized a heuristic approach to optimize flip-flops in SoCs, considering aging, voltage, and Bias Temperature Instability (BTI) variations, aiming to improve circuit timing and reliability. The limitation of this method is the use of a first-order approximation of the delay to perform the optimization. Authors in [8] proposed a transistor sizing approach for portable electrical circuits using flexible thin-film transistors (TFTs), combining fresh and lifetime yields into a single objective function for optimization. This method combines fresh and lifetime yields into a single objective function to optimize. In [24], a new BO was presented to approximate variations using a Gaussian kernel and solve the optimization problem using a multi-start-point (MSP) expectation-maximization (EM)-like approach. This would result in a waste of computational resources for low-yield candidates. In [25], a PSO-based optimization

algorithm was proposed to determine the optimal representation of Open Circuit Voltage (OCV) through polynomial regression. In [26] an Adaptive Yield Estimation based on BO (ABO) was proposed. This approach aims to decrease the number of samples required to run the Bayesian algorithm in various scenarios. Moreover, it utilizes the weighted expected improvement (WEI) acquisition function to achieve a better point in the design space of the problem while considering the constraints. In [27], a statistical design approach for near-threshold specialized circuits was developed, to improve performance and reduce area. In [28], an analytical formula for yield optimization of Full adders was derived, considering Negative Bias Temperature Instability (NBTI) and PVT variations. One of the limitations of this method is that complex performance metrics such as timing jitter don't have closed-form expressions. In [29], a modified generalized boundary curve (GBC) algorithm was utilized to address CNFET variations in CNFET-based circuits. Authors in [30] applied a Residual Neural Network as a surrogate model for Genetic algorithm (GA) in circuit yield optimization. A FlexiOptimizer interface for yield maximization of TFT was presented in [31], using an orthogonal array (OA) method to consider aging and bending effects. A cross-layer framework and analytical approach for designing and optimizing FinFet SRAMs under process variation were introduced in [32].

A process-aware surrogate model of MOSFET physics and circuit performance metrics using Gaussian process regression was generated in [33], which was then used for simulation-based multi-objective optimization to increase circuit yield while satisfying design goals. A yield optimization method that combines fuzzy c-means (FCM) and an evolutionary population-based algorithm was proposed in [34], requiring fewer iterations than MC. However, it suffers from clustering problems. In [35], the Fast Sensitivity Importance Sampling (FSIS) method was proposed as an inner yield analysis technique. It uses finite differences in transient sensitivity analysis and multiple simulations to identify the most probable failure point (MPFP). A linear surrogate model is then employed to model the constraint space, speeding up the optimization process. Finally, a yield function is defined for outer optimization, calculating new points through finite difference and passing them to inner yield analysis to estimate the yield. In [36], an improved version of the all-sensitivity adversarial importance sampling (ASAIS) for yield maximization is presented. This version is generalized into an effective proposal distribution transfer (OPT) procedure, captured using conditional normalizing flow (CNF). In [37], a novel yield estimation method called OPTIMIS is introduced, which combines surrogate-based and importance sampling techniques.

The reviewed methodologies, including those in [1], [3], [7], [14], and [17], exhibit limitations such as oversimplification or approximation of primary issues, leading to reduced accuracy. Additionally, some methodologies, like those in [16], [20], and [28], are designed for specific

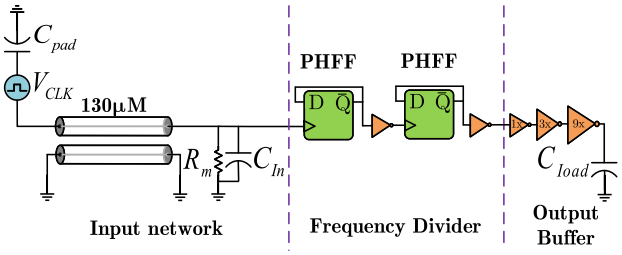


FIGURE 1. The output buffer with PHFF-based frequency divider and connected transmission line schematic.

circuits and are not universally applicable to all circuit design challenges or performance metrics. State-of-the-art simulation-based methods for yield maximization, such as those in [7] and [34], often consider manufacturing process variations but pose significant challenges due to their time-consuming nature. While surrogate models, as seen in [21], [24], and [25], have shown promise in expediting this process, they cannot guarantee the attainment of the global optimal point. Furthermore, mathematical-based models for yield maximization, such as those in [5], may suffer from reduced reliability due to the complex nature of circuit performance metrics or the lack of available mathematical relationships.

A. THE PHFF AND LLTFF FLIP-FLOPS

To address these issues, researchers are actively working to achieve faster yield optimization processes with reliable accuracy, particularly by leveraging state-of-the-art computer-aided design (CAD) techniques. This paper proposes a solution to these challenges using the deep multi-layer perceptron neural network (DNN), a powerful CAD method capable of accurately modeling circuit performance metrics based on the universal approximation theorem [38]. Our approach involves DNN-based yield maximization, which provides accurate closed-form formulations for performance metrics and is assisted by the analytical form of circuit manufacturing yield, enabling powerful gradient-based optimization.

The remaining parts of this paper is structured as follows: section II presents the concept of our proposed methods, starting with the mathematical model for yield, which replaces the MC simulation. Also, the polyhedral approximation of the nonlinear constraints is presented. Then, the DNN structure and data generation process are introduced to generate the DNN-based models of the performance metrics. The yield maximization results and discussions for the PHFF and LLTFF flip-flops are presented in section III. Finally, section V concludes the paper.

II. PROPOSED YIELD MAXIMIZATION METHODS

In this section, we present a mathematical formulation for yield maximization, incorporating the DNN-based surrogate model. This novel approach enables the utilization of

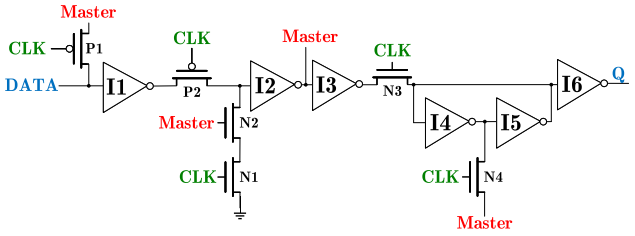


FIGURE 2. The transistor-level schematic of the PHFF with its input signals and transistors name.

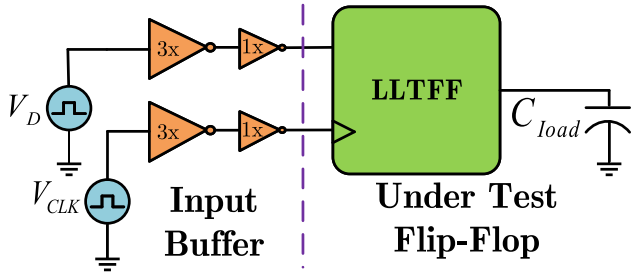


FIGURE 3. LLTFF with multistage input receiver schematic.

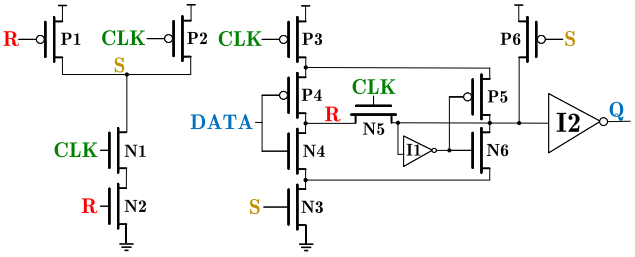


FIGURE 4. The transistor-level schematic of the LLTFF with its input signals and transistors name.

fast and powerful gradient-based optimization methods, ensuring efficient and accurate convergence without relying solely on time-consuming MC simulations. To validate the effectiveness of our proposed method, we select two state-of-the-art flip-flop structures as case studies. These structures serve as benchmarks for evaluating the performance of our approach. Furthermore, we introduce the formulations and fundamental concepts of the polyhedral-approximated yield maximization method (PA yield). Subsequently, we outline the process of our DNN-based yield maximization method, leveraging DNNs to model and predict circuit behavior. We provide a detailed explanation of how the DNN-based modeling process utilizes training data generated from the transistor-level models existing in the Spice simulator. The number of training data required is determined experimentally. By adopting this approach, we aim to enhance the efficiency and accuracy of yield maximization in circuit design.

The proposed DNN-based yield maximization method is validated by implementing two selected flip-flops in 16 nm PTM-CMOS technology. In the first example (Fig. 1), obtained from [39], a clock generator source drives a

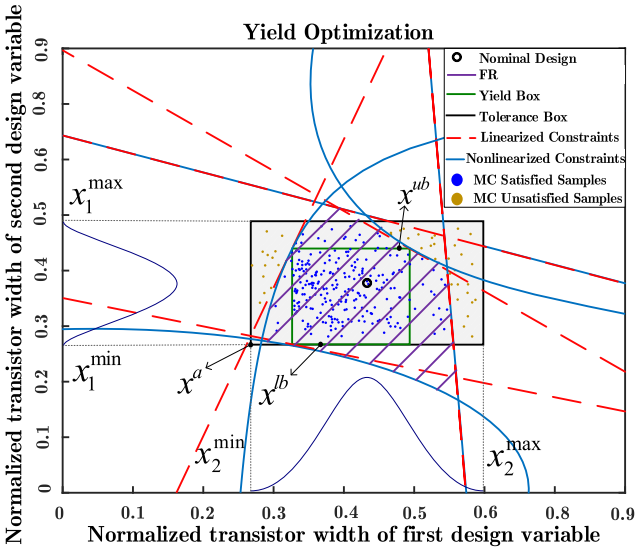


FIGURE 5. Simplified two-dimensional yield optimization process for two design variables x_1 and x_2 with their nominal point.

transmission line connected to the input of a frequency divider (divided by 4). The output of the frequency divider is then passed to a buffer to drive an output load. We replaced the flip-flop presented in [39] with the one in [12], resulting in a more complex circuit for practical demonstration. The schematic of the PHFF from [12] is shown in Fig. 2. The second example, as presented in [9], involves the LLTFF with a two-stage receiver (Fig. 3). The SPICE transistor-level structure of the LLTFF is depicted in Fig. 4 [9]. The primary performance metrics considered in the DNN-based yield maximization process include dynamic power (*DP*), setup time (*ST*), hold time (*HT*), data to Q delay (*DQ*), and clock to Q delay (*CQ*) [9], [12].

A. FORMULATION OF STATISTICAL DESIGN

This section introduces a statistical design approach to maximize the yield of PHFF and LLTFF circuits while accounting for process variation. It is assumed that the PDF (probability distribution function) of each design variable and constraint in these circuits follows a Gaussian distribution (GD) with a three-sigma ($\pm 3\sigma$) variation [1], [11]. The upper bounds (*UB*) for the design constraints are considered as μ_{DQ} for DQ delay, μ_{CQ} for CQ delay, μ_{ST} for setup time, μ_{HT} for hold time, and μ_{DP} for dynamic power. Thus, the circuit performance metrics are transformed into the following statistical design constraints to accommodate process variation:

$$\begin{cases} DQ + 3(\sigma_{DQ}) \leq \mu_{DQ}, \\ CQ + 3(\sigma_{CQ}) \leq \mu_{CQ}, \\ ST + 3(\sigma_{ST}) \leq \mu_{ST}, \\ HT + 3(\sigma_{HT}) \leq \mu_{HT}, \\ DP + 3(\sigma_{DP}) \leq \mu_{DP}. \end{cases} \tag{1}$$

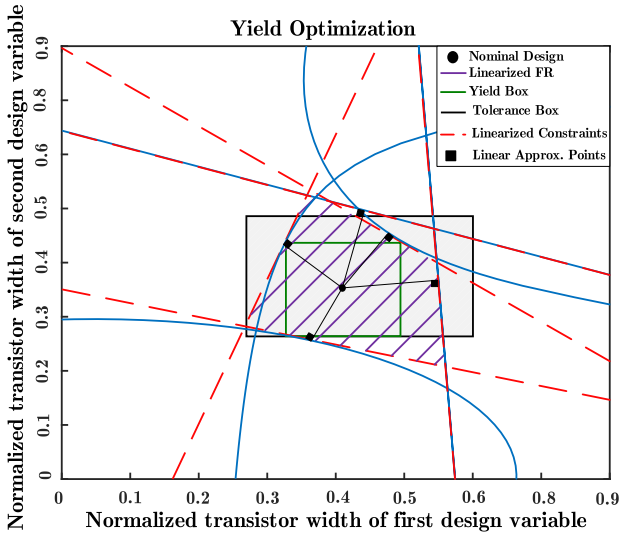


FIGURE 6. Representation of the finding closest point to the linearized constraint by minimizing equation (5).

in equation (1), a three-sigma variation is added to the left-hand side of each constraint [40]. By applying the design constraints described in equation (1) to the solution space, a feasible region (FR) is determined, ensuring the satisfaction of all design constraints. It should be emphasized that the nominal point of the FR for initiating yield maximization is obtained from a deterministic design without considering process variation. Fig. 5 illustrates a simplified example of yield optimization for a two-dimensional problem, including two normalized design variables (transistor widths) and nonlinear constraints. In this figure, the tolerance box is defined as a box centered at the nominal point and expanding by a tolerance percentage ($\pm 3\sigma$) in both dimensions. Consequently, the size of the tolerance box is directly influenced by the distribution of the design variables. Based on equation (1), the FR depicted in Fig. 5 encompasses all points within the tolerance box that satisfy the nonlinearized constraints (represented by blue dots), which indicates the overlapping region between the FR and the tolerance box. To optimize yield, it is necessary to shift the nominal point along with its corresponding tolerance box to achieve maximum overlap with the FR. An optimization algorithm is employed due to the complexity of determining the overlapping areas. In Fig. 5, the approximated yield is represented by the largest rectangle (yield box) obtained from the overlap between the FR and the tolerance box [40]. It is important to note that for the PHFF and LLTFF examples discussed in this paper, the solution space is a multi-dimensional space with constraints different from the two-dimensional space depicted in Fig. 5.

1) YIELD MAXIMIZATION USING POLYHEDRAL APPROXIMATION OF NONLINEAR CONSTRAINTS

In this section, the constraints are linearized, resulting in a polyhedral estimation of the FR. The performance metrics constraints space defines the FR as follows [40]:

$$FR = \{x \in D^r | C_n(x) \geq 0, \qquad n = 1, 2, \ldots, N\} \quad (2)$$

where $x \in X = [x_1, \ldots, x_r]$ and represents the vector of the random design variables with an arbitrary joint probability density function (PDF). Also, D^r is the r dimensional design space, r is the number of design variables, $C_n(x)$ is the n^{th} nonlinear constraints and N indicates the number of constraints. The term $C_n(x) \geq 0$ reflects the n^{th} constraint to be satisfied. However, in many cases, obtaining the precise mathematical closed-form expression for $C(x)$ is either unknown or available with limited accuracy, and it is often feasible to only obtain the gradient of the function through numerical evaluation. To estimate the polyhedral shape of the FR, a polyhedral estimation technique can be employed by iterative computing with new nominal points. This approach utilizes the first-order Taylor series estimation of any $C_n(x)$ to estimate the polyhedral shape of the FR. So the derivatives of the performance metrics to the design variables x are presented as follows [40]:

$$\frac{\partial(DQ(x))}{\partial(x)}, \frac{\partial(CQ(x))}{\partial(x)}, \frac{\partial(ST(x))}{\partial(x)},$$

$$\frac{\partial(HT(x)}{\partial(x)}, \frac{\partial(DP(x))}{\partial(x)} \quad (3)$$

the first-order estimation of each performance metrics constraints $C_n(x)$ at x^* point can be calculated as follows [40]:

$$C_{ln,n}(x) \approx C_n(x^*) + g_n(x^*)^T . (x - x^*) \quad (4)$$

where $C_{ln,n}(x)$ refers to the n^{th} linearized constraint named polytope in Fig. 6 and $g_n(x^*)$ is the derivative vector of $C_n(x^*)$. This shared region between the polyhedral approximation of the nonlinear constraint region by finding x^* point for each constraint creates a new FR in Fig. 6 [40]. By solving the following optimization problem, we can determine the x^* point for each constraint, which is the point on the $C_n(x) = 0$ surface that has the minimum distance to the middle of the tolerance box [40]:

$$\begin{aligned} &min \quad \Delta = \sqrt{(x - x^m)^T . (x - x^m)} \\ &s.t \quad C_n(x) = 0 \end{aligned} \quad (5)$$

where x^m refers to the middle point of the tolerance box or nominal design. In summary, Figure 5 illustrates two design variables, x_1 and x_2, each exhibiting a three-sigma variation within the design space. The feasible region is visually represented by blue and yellow points, indicating the interior and exterior of this region, respectively. The modification of x_1 and x_2 values directly impacts the circuit's performance metrics and constraints, such as power and delay, and influences the design variables, as shown by the blue lines in Figure 5. The proposed approach aims to identify the largest polyhedron within the collective feasible region, thereby maximizing manufacturing yield. The graphical representation in Figure 6 illustrates the

iterative process involved in the search for a design point that optimizes yield. This iterative procedure focuses on determining the minimum distance to the linear constraints that are perpendicular to the design point. In traditional design, symmetrical distributions are commonly used for simplicity. In such cases, the maximum yield box, as shown in Fig. 5 (smaller box), represents the maximum yield in the FR. However, when the probability density function (PDF) is non-symmetrical, this box no longer accurately represents the maximum yield. Calculating the yield in such cases requires more complex techniques, such as multi-dimensional probability integrals or MC simulations, which can be computationally intensive [41]. To address this issue, the proposed approach utilizes Kumaraswamy's distribution for estimating the double-bounded (DB) PDF of physically bounded variables as follows [40]:

$$u(s) = q.\,p.\,s^{q-1}.(1-s^q)^{p-1}$$

and

$$s = \frac{x - x^{min}}{x^{max} - x^{min}}, \quad x^{min} \le x \le x^{max} \tag{6}$$

here, $u(s)$ represents Kumaraswamy's distribution function, and x^{max} and x^{min} refer to the upper and lower bounds of the variable x in Fig. 5. The shape of the DB-PDF can be adjusted by changing the values of q and p. It is worth noting that the cumulative distribution function (CDF) of (6) can be computed as follows [40]:

$$U(s) = 1 - (1 - s^q)^p \tag{7}$$

where U refers to the integral of Kumaraswamy's distribution.

2) YIELD MAXIMIZATION

Assuming a convex polyhedral feasible region (PFR), the goal of yield maximization is to find the maximum yield rectangular multi-dimensional cube (D) that lies inside the PFR [40]. This cube is defined as:

$$D(x^{lb}, x^{ub}) = \{x \in D^r \mid \quad x^{lb} \le x \le x^{ub}\} \tag{8}$$

where x^{ub} and x^{lb} represent the upper and lower bounds of the yield box ($D(x^{lb}, x^{ub})$) in Fig. 5 and r is the number of design variables. Inside the PFR, the $D \subseteq PFR$ is analogous to:

$$L^+.x^{ub} - L^-.x^{lb} \ge K \tag{9}$$

the L^- and L^+ refer to the lower and upper bounds of the selected constraint, respectively. K refers to the constant value of the RHS of the linearization. Additionally, considering a tolerance box that contains the middle point (x^m) and its corresponding tolerance percentage (t), a multi-dimensional polyhedral shape can be defined as follows [40]:

$$[x^m - \frac{t}{2},\ x^m + \frac{t}{2}] = [x^{min}, x^{max}] \tag{10}$$

where, the tolerance box can be represented by design variables vector, which is varied by $t/2$ in both directions and also:

$$x^{max} - x^{min} = t = 6 \times \sigma_x \tag{11}$$

according to the previous definitions, the yield function can be defined as follows:

$$\begin{aligned} Yield(x^a, x^{ub}, x^{lb}) &= \prod_{v=1}^{r} P\{x_v^{lb} \le x_v \le x_v^{ub}\} \\ &= \prod_{v=1}^{r} \left[U\left(\frac{x_v^{ub} - x_v^a}{t_v}\right) - U\left(\frac{x_v^{lb} - x_v^a}{t_v}\right) \right] \end{aligned} \tag{12}$$

where x^a refers to the bottom-left corner of the yield box in Fig. 5. Also, x and t indicate the r^{th} design variable and its corresponding tolerance percentage. Finally, the yield maximization problem can be formulated as follows, considering the objective function and all constraints as closed-form mathematical formulas:

$$\begin{aligned} max \quad & Yield(x^a, x^{ub}, x^{lb}) \\ s.t \quad & \begin{cases} L^+.x^{ub} - L^-.x^{lb} \ge K, \\ x^a \ge x^{min}, \\ x^{lb} \ge x^a, \\ x^{ub} - x^{lb} \ge t, \\ x^a + t \le x^{max}. \end{cases} \end{aligned} \tag{13}$$

in this subsection, the approach for circuit yield maximization is explained, focusing on approximating the constraints space with a limited approximation error. However, due to the absence of precise analytical formulas for the performance metrics in this method, it becomes necessary to utilize the simulator in certain steps. As a result, in the following section, we present the proposed DNN-based yield maximization method, which aims to enhance the efficiency of the yield maximization process using yield analytical formula and nonlinear neural network models instead of linear approximation.

B. THE PHFF AND LLTFF PERFORMANCE METRICS MODELING FOR USE IN THE DNN-BASED YIELD MAXIMIZATION

In this section, the paper describes the modeling process necessary to incorporate the closed-form formula of the neural network into the proposed yield maximization method. We will discuss the data generation process, the structure of the neural network, and the training process required to obtain this model. It is worth noting that the DNN-based models provide faster evaluation times than the transistor-level models existing in the Spice simulator and higher accuracy than analytical models [7], [42].

TABLE 1. Training/testing errors percentage and cpu time comparison between the DNN-Based and the transistor-level models existing in the Spice simulator.

Circuit example	Performance metric	Training error percentage	Testing error percentage	Transistor-level model simulation time (ms)	DNN-based model simulation time (ms)	Speedup ratio
PHFF	DQ delay	0.26	0.30	1282	3.63	352
	CQ delay	0.08	0.09	1350	4.14	326
	Setup time	0.75	0.80	1265	4.20	301
	Hold time	0.98	0.99	1336	4.89	273
	Dynamic power	0.60	0.64	1210	4.52	267
LLTFF	DQ delay	0.50	0.52	103	4.52	23
	CQ delay	0.72	0.77	105	5.22	20
	Setup time	0.38	0.39	100	5.68	17
	Hold time	0.12	0.13	107	5.76	18
	Dynamic power	0.18	0.20	110	5.56	19

1) DATA GENERATION

Data generation serves as the initial step in creating the DNN-based models to evaluate the performance metrics of flip-flops. It is worth noting that the DNNs with enough hidden neurons can learn any nonlinear input-output relationship of a circuit with a desired level of accuracy by generating enough training data according to the universal approximation theorem [38].

In the proposed DNN-based yield maximization problem, the widths of circuit transistors are designated as the design variables. For the PHFF circuit (Fig. 2), the design variables chosen for swiping and data generation encompass the $P1$ and $P2$ transistors, along with the range of $N1$ to $N4$ transistors. These specific design variables were selected due to their significant impact on the performance metrics of the PHFF circuit. Similarly, in the case of the LLTFF example, the design variables chosen for swiping and data generation consist of the $[P1, P3, P4]$ transistors, as well as the $[N2 : N4]$ transistors depicted in Fig. 4. These particular design variables are deemed critical as they substantially influence the performance metrics of the LLTFF circuit. To ensure accuracy in training the DNN-based models, transistor-level models existing in the Spice simulator were employed for data generation. These models offer the necessary level of precision, allowing for an accurate representation and understanding of the circuit's behavior by the DNN.

2) NEURAL NETWORK STRUCTURE

The structure of the DNN is composed of layers, neurons, and connections. Each layer can have a different number of neurons and can employ various activation functions, such as linear, sigmoid, exponential, and hyperbolic tangent [43]. Fig. 7 illustrates that every DNN consists of at least two hidden layers, an input layer, and an output layer. In this figure, x_i represents the i^{th} input of the DNN, l indicates the number of the DNN layer from $[1 : L]$, n_l is the number of neurons in layer l, b_i^l is the bias of i^{th} neuron of layer l, w_{ij}^l is the weight connecting i^{th} neuron of l^{th} layer to the j^{th} neuron of $(l-1)^{th}$ layer. In this figure, a_j^l is the input of the activation function of j^{th} neuron in layer l, and o_j^l refers to the final output of the same neuron. So, the output of the DNN

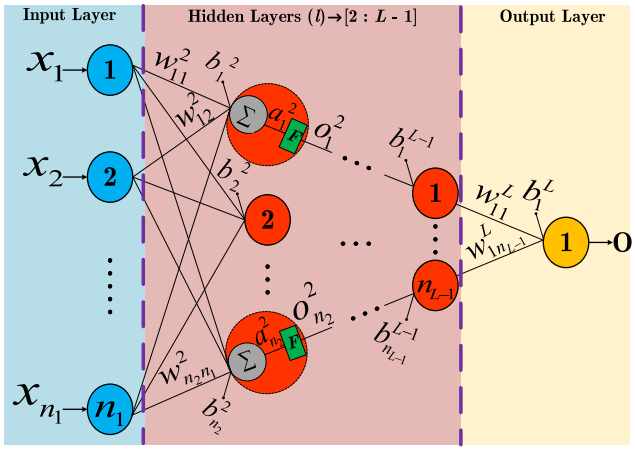


FIGURE 7. Deep multi-layer perceptron neural network schematic with multi-inputs and single output.

can be computed as follows:

$$o = F(\sum_{j=1}^{n_{L-1}} w_{1j}^L \times o_j^{L-1}) \tag{14}$$

where F refers to the activation function, and o_j^{L-1} is the output of j^{th} neuron of layer $L-1$. The accuracy of the DNN model is influenced by the choice of training and testing data adopted during the training process. The training process of the DNN revolves around minimizing a predefined error function through the adjustment of weights and biases. The Mean Square Error (MSE) is a widely employed error function, defined as follows:

$$Er_{(Total)} = 0.5 \sum_{h \in N_s} \left| F(x_h, W) - o_h^d \right|^2 \tag{15}$$

where $Er_{(Total)}$ is the training error function, N_s presents the number of the training samples, $\boldsymbol{W}$ indicates the matrix of the weights of the DNN, x_h is the h^{th} training data, $F(x_h, W)$ indicates the DNN output for h^{th} training data, and o_h^d is the desired output for h^{th} training data.

Stopping criteria can adjust the DNN model's accuracy. By setting the stopping criteria to a high (99%) accuracy, the

TABLE 2. DNN structure used for modeling performance metrics.

Circuit example	Performance metric	Structure	No. of DNN parameters
PHFF	DQ delay	$6, 8, 5, 3, 1$	123
	CQ delay	$6, 8, 5, 3, 1$	123
	Setup time	$6, 8, 5, 5, 1$	137
	Hold time	$6, 15, 10, 8, 1$	362
	Dynamic power	$6, 15, 10, 5, 1$	326
LLTFF	DQ delay	$6, 15, 10, 5, 1$	326
	CQ delay	$6, 10, 5, 3, 1$	147
	Setup time	$6, 12, 8, 5, 1$	239
	Hold Time	$6, 15, 12, 10, 1$	438
	Dynamic power	$6, 10, 7, 4, 1$	214

DNN models and subsequently final yield can be as accurate as the original.

3) TRAINING OF THE DNN FOR GENERATION MODELS OF PERFORMANCE METRICS OF THE FLIP-FLOPS

The modeling and yield maximization process for flip-flops can be achieved using the DNN-based training method. In this approach, five performance metrics, namely DP, ST, HT, DQ, and CQ, were selected for modeling and used in the yield maximization process. Table 1 presents the testing and training error percentages of the DNN-based models for each performance metric in both examples. Here, the error percentage is used instead of the mean square error to measure accuracy, providing a better comprehension of the results of the neural network model. It also provides a comparison between the transistor-level models existing in the Spice simulator and the DNN-based models, showing a significant speedup ratio for the DNN-based models. Furthermore, the DNN-based models exhibit high accuracy compared with the transistor-level models existing in the Spice simulator across all performance metrics in both examples. This suggests that these fast and accurate models can effectively replace the time-consuming transistor-level models existing in the Spice simulator. By utilizing these fast and accurate DNN-based models, the optimization process can be significantly accelerated compared with the heuristic/simulation-based optimizations. This enables a much faster and more efficient optimization process.

Table 2 presents the DNN structures for each performance metric and the total number of parameters (weights and biases) used in the DNN-based models. We first started with a small, shallow neural network containing only a few hidden neurons and trained it. We then gradually increased the number of neurons in each layer up to a certain limit (e.g., 15). If the network still did not meet the desired training error, we added another layer. This process was repeated until the network achieved the desired level of accuracy. Our focus was on achieving high accuracy during both training and testing while ensuring that the network generalizes well within the feasible region. These models employ three hidden layers with either sigmoid or hyperbolic tangent activation

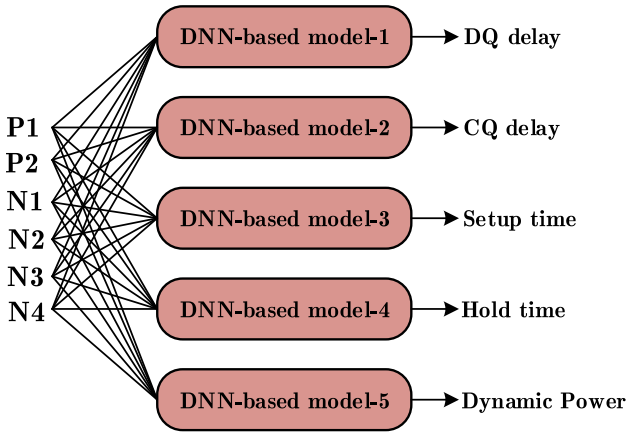


FIGURE 8. Trained DNN-based model of the PHFF performance metrics which receives the width of the transistors at the input and generates the desired performance metrics.

functions. In this study, different neural network structures were used to address under-learning issues and minimize the difference between test and training errors to reduce overfitting. Using a single neural network for all performance metrics would lead to some metrics being overfitted while others remain underfitted, making efficient training more difficult. Therefore, different neural networks were used for different performance metrics to ensure that the error of each performance measure was within an acceptable range. The selected performance metrics were modeled with a few DNN parameters to minimize the probability of overfitting. Fig. 8 illustrates the generated model structures for different performance metrics in the PHFF examples. In Fig. 8, the inputs $[P1, P2]$ and $[N1 : N4]$ refer to the NMOS and PMOS transistor widths of the PHFF example. For the LLTFF example, the performance metrics are modeled similarly. In this case, the widths of the $[P2, P3, P4]$ and $[N2 : N4]$ transistors are chosen as design variables. The transistors selected for optimization in this study have been determined to significantly impact the performance metric values of the PHFF and LLTFF circuits. Each DNN-based model represents a closed-form feedforward neural network mathematical formula. This formula takes transistor widths as inputs and generates the corresponding output for each performance metric shown in Fig. 8.

4) DNN-BASED DETERMINISTIC DESIGN

Identifying an appropriate nominal point (the deterministic design) is a crucial step in initiating yield optimization. DNN-based optimization is employed to accomplish this, wherein one performance metric is designated as the objective function while the remaining performance metrics are defined as constraints. The purpose of this optimization is to minimize the objective function while simultaneously ensuring the satisfaction of all constraints as follows:

$$min \quad DP_{DNN}(x_z) =$$

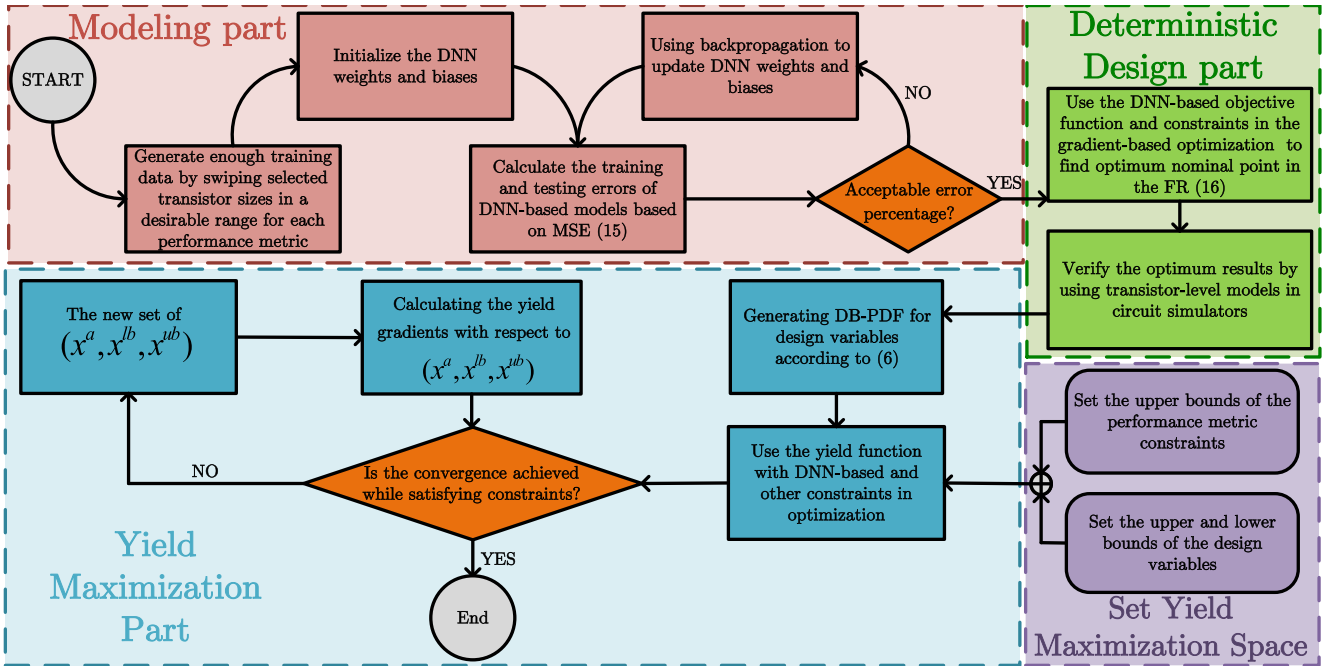


FIGURE 9. Flowchart of the proposed DNN-based yield maximization technique which includes both deterministic and probabilistic design.

$$\left(\sum_{j=1}^{5}\sigma\left(\sum_{k=1}^{10}\sigma\left(\sum_{i=1}^{15}\sigma\left(\sum_{z=1}^{6}w_{iz}^{1}.x_z+b_i^1\right)\right.\right.\right.$$
$$\left.\left.\left..w_{ki}^2+b_k^2\right).w_{jk}^3+b_j^3\right).w_{1j}^4+b_1^4\right)$$
$$s.t\quad\begin{cases}DQ_{DNN}(x_z)\le\mu_{DQ},\\ CQ_{DNN}(x_z)\le\mu_{CQ},\\ ST_{DNN}(x_z)\le\mu_{ST},\\ HT_{DNN}(x_z)\le\mu_{HT}.\end{cases}\tag{16}$$

where the μ_{DQ}, μ_{CQ}, μ_{ST}, and μ_{HT} are right-hand side (RHS) constant values referring to the upper bounds of DQ delay, CQ delay, setup time, and hold time constraints, respectively. Also, DP_{DNN} is the DNN-based closed-form formula for dynamic power. Furthermore, σ refers to the sigmoid activation function. Additionally, x_z denotes the z^{th} design variable in Fig. 8, where x_z ranges from one to six. The gradient of the objective function with respect to the transistor width is computed as follows:

$$\begin{aligned}\frac{\partial DP_{DNN}}{\partial x_z}&=\frac{\partial o_1^4}{\partial x_z}\\&=\frac{\partial o_1^4}{\partial a_1^4}.\frac{\partial a_1^4}{\partial o_j^3}.\frac{\partial o_j^3}{\partial a_j^3}.\frac{\partial a_j^3}{\partial o_k^2}.\frac{\partial o_k^2}{\partial a_k^2}.\frac{\partial a_k^2}{\partial o_i^1}.\frac{\partial o_i^1}{\partial a_i^1}.\frac{\partial a_i^1}{\partial x_z}\\&\quad+\frac{\partial o_1^4}{\partial a_1^4}.\frac{\partial a_1^4}{\partial o_{j+1}^3}.\frac{\partial o_{j+1}^3}{\partial a_{j+1}^3}.\frac{\partial a_j^3}{\partial o_k^2}.\frac{\partial o_k^2}{\partial a_k^2}.\frac{\partial a_k^2}{\partial o_i^1}.\frac{\partial o_i^1}{\partial a_i^1}.\frac{\partial a_i^1}{\partial x_z}\\&\quad+\ldots.\\&=\left(w_{1j}^4.\sigma\prime(o_j^3).w_{jk}^3.\sigma\prime(o_k^2).w_{ki}^2.\sigma\prime(o_i^1).w_{iz}^1\right)\\&\quad+\left(w_{1j+1}^4.\sigma\prime(o_{j+1}^3).w_{j+1k}^3.\sigma\prime(o_k^2).w_{ki}^2.\sigma\prime(o_i^1).w_{iz}^1\right)\\&\quad+\ldots.\\&\qquad for\quad 1\le z\le 6\end{aligned}\tag{17}$$

5) YIELD MAXIMIZATION USING DNN-BASED NONLINEAR CONSTRAINTS

In the previous subsection, the polyhedral approximation approach was used to approximate the constraints space when closed-form formulas for the constraints were not available. However, in the proposed method, a different approach was employed. Instead of relying on a polyhedral approximation, a neural network surrogate model is utilized to derive a closed-form formula for the constraints. Using a neural network surrogate model, the original constraints can be modeled and represented by a mathematical formula. This eliminates the need for approximation and provides a more accurate representation of the constraints and a much faster yield maximization process. As a result, the error in maximizing the yield is substantially reduced compared to the polyhedral approximation approach.

The proposed DNN-based yield maximization method utilizes the upper bounds of the yield maximization, as indicated in equation (18), which is derived from the closed-form formula obtained through the neural network. This approach replaces the linearized constraints used in previous methods with the more accurate closed-form formula obtained from the neural network.

$$max\quad Yield(x^a, x^{ub}, x^{lb})$$

TABLE 3. Upper and lower bounds of the PHFF design variables.

Variable	P1	P2	N1	N2	N3	N4
UBs (m)	$37e-9$	$37e-9$	$22e-9$	$22e-9$	$85e-9$	$88e-9$
LBs (m)	$27e-9$	$27e-9$	$16e-9$	$16e-9$	$76e-9$	$70e-9$

$$s.t \quad \begin{cases} \text{Original nonlinear constraints from (1)}, \\ x^a \geq x^{min}, \\ x^{lb} \geq x^a, \\ x^{ub} - x^{lb} \geq t, \\ x^a + t \leq x^{max}. \end{cases} \tag{18}$$

The proposed methodology employs a rapid analytical approach to yield estimation, eliminating the need for Monte Carlo simulation. This method leverages DNN-based models of circuit performance metrics, integrating their closed-form mathematical expressions into the analytical yield formula and optimizing them using a gradient-based approach. Notably, DNN-based yield maximization can encompass a wide range of design variables that affect manufacturing yield, with their impact on performance metrics being assessable through simulation or measurement.

Fig. 9 illustrates the flowchart of the proposed method, which consists of three main parts: performance metrics DNN-based modeling, deterministic design, and DNN-based yield maximization. By implementing this DNN-based optimization technique, transistor designs can achieve maximum yield under process variation, resulting in highly efficient circuit designs. Moreover, this approach is flexible and can be applied to various circuits and fabrication technologies, providing a versatile solution for yield maximization. Using the DNN-based model has resulted in a significantly more precise approximation than imprecise analytical formulas.

III. YIELD MAXIMIZATION RESULTS

The proposed manufacturing yield maximizing method is demonstrated by executing various scenarios on both circuit examples in this section. The first step in ensuring circuit performance meets the required metrics is to identify a suitable nominal point within the FR. For each circuit, this is achieved by using the DNN-based constrained optimization while disregarding the constraints sigma in equations (19) and (20). In other words, the nominal point is a deterministic design that does not account for any process variations. To address this, DNN-based models were utilized and passed into a gradient-based optimizer to find the optimal solution for both examples. The statistical design results of the two examples are examined separately to evaluate the effectiveness of the proposed method.

A. PHFF EXAMPLE RESULT

Table 3 provides the upper and lower limits for the design variables and performance metrics associated with the PHFF example. These bounds are crucial for effectively defining the problem space. The upper bound of constraints is determined by utilizing the average value of performance metric data generated in equation (19).

$$\begin{cases} DQ_{DNN} + 3(\sigma_{DQ_{DNN}}) \leq 84e-12\ (s), \\ CQ_{DNN} + 3(\sigma_{CQ_{DNN}}) \leq 57e-12\ (s), \\ ST_{DNN} + 3(\sigma_{ST_{DNN}}) \leq 33e-12\ (s), \\ HT_{DNN} + 3(\sigma_{HT_{DNN}}) \leq 18e-12\ (s), \\ DP_{DNN} + 3(\sigma_{DP_{DNN}}) \leq 4e-06\ (w). \end{cases} \tag{19}$$

This helps in finding a correct point within the FR. The nominal point of the PHFF is obtained from the DNN-based optimization presented in equation (16) and detailed in Table 4. Unfortunately, the PHFF example achieved a yield of only 71.43%, indicating suboptimal performance. So, yield maximization methods should be employed to reduce manufacturing costs.

Tables 5 and 6 provide comparisons between the proposed PA yield and DNN-based yield maximization methods and three other algorithms: simulation-based particle swarm optimization (SM-PSO), simulation-based PSO assisted with polynomial regression as an online surrogate model (SM-PSO-PR) [25] and ABO [26]. Table 5 presents the final yield values, design variables, and relevant performance metrics. The DNN-based method stands out for its ability to effectively converge to optimal design variables with a high yield. This success can be attributed to the powerful combination of a fast and accurate DNN-based model with mathematical yield maximization using a strong gradient-based optimizer. The DNN-based approach effectively determines the optimal transistor sizes that maximize the yield. However, it may not necessarily achieve the best possible values for other performance metrics when compared to the PA, SM-PSO, SM-PSO-PR, and ABO yield maximization methods.

It is worth noting that the main objective of the optimization is to maximize yield, with less emphasis on minimizing other performance metrics. However, as demonstrated in Table 5, achieving a higher yield generally corresponds to lower values in other performance metrics. Since the proposed PA yield and DNN-based yield maximization methods, the SM-PSO-PR method, and ABO employ surrogate models and approximations, it is crucial to verify their results using the MC method. Table 6 provides the error percentages of these methods compared to the MC simulations, utilizing transistor-level models existing in the Spice simulator in the circuit simulator. The error percentages are determined by comparing the MC simulation results with the predicted outcomes.

The DNN-based yield maximization method exhibits higher accuracy compared to the PA yield method due to several factors. Firstly, the DNN-based method does not involve an approximation process, unlike the PA yield method which requires solving optimization and approximation equations (4) and (5) respectively. This allows the DNN-based method to utilize the original DNN-based constraints model instead of an approximated one, leading to improved accuracy. Additionally, the DNN-based yield maximization

TABLE 4. Obtained nominal point from equation (16) for PHFF example.

Optimum design variables	Performance metric array					Yield
$[P1, P2, N1, N2, N3, N4](m)$	$DQ(s)$	$CQ(s)$	$ST(s)$	$HT(s)$	$DP(w)$	
$[27, 37, 16, 16, 76, 88]e-09$	$82.26e-12$	$55.99e-12$	$26.29e-12$	$17.44e-12$	$3.75e-06$	71.43

TABLE 5. Comparison of the yield percentage obtained by the proposed DNN-Based, the PA Yield, the SM-PSO, and the SM-PSO-PR yield maximization methods for a PHFF example.

Yield maximization methods	Optimum design variables	Optimum performance metric array					Yield
	$[P1, P2, N1, N2, N3, N4](m)$	$DQ(s)$	$CQ(s)$	$ST(s)$	$HT(s)$	$DP(w)$	
SM-PSO	$[36, 30, 17, 16, 77, 72]e-09$	$82.74e-12$	$56.24e-12$	$26.42e-12$	$17.70e-12$	$3.88e-06$	96.35
SM-PSO-PR [25]	$[35, 29, 16, 20, 76, 73]e-09$	$81.06e-12$	$56.48e-12$	$25.75e-12$	$17.56e-12$	$3.84e-06$	97.91
ABO [26]	$[30, 34, 20, 17, 78, 78]e-09$	$82.58e-12$	$56.01e-12$	$26.57e-12$	$17.52e-12$	$3.84e-06$	99.61
PA yield	$[27, 34, 17, 20, 78, 72]e-09$	$82.77e-12$	$56.03e-12$	$26.75e-12$	$17.74e-12$	$3.86e-06$	96.50
Proposed DNN-based yield	$[30, 34, 16, 19, 76, 74]e-09$	$82.42e-12$	$56.00e-12$	$26.42e-12$	$17.45e-12$	$3.83e-06$	100

TABLE 6. Comparison of the Iterations, CPU Time, Speedup, and Accuracy of the Proposed DNN-Based, the PA Yield, the SM-PSO, and the SM-PSO-PR yield maximization methods for a PHFF example.

Yield maximization methods	Number of iterations	Total time (min)	Error percentage	Yield based on Table 5	Yield based on MC	Speed up
SM-PSO	150	4020	–	96.35	96.35	1
SM-PSO-PR [25]	210	264	2.61	97.91	95.30	15
ABO [26]	80	107	0.47	99.61	99.14	38
PA yield	53	18	7.31	96.50	89.31	223
Proposed DNN-based yield	32	1.1	0.10	100	99.90	3654

method benefits from the universal approximation theory, which enables it to train on sufficient data and achieve a highly precise model. As a result, the DNN-based method demonstrates a significantly lower error percentage compared to the SM-PSO-PR method and comes close to the results obtained from simulator software.

BO's method uses the Gaussian process regression model to estimate the yield value, requiring updates with new data at each iteration. This approach has successfully achieved a suitable level of accuracy. However, leveraging the universal approximation capability of the proposed DNN-based yield maximization method allows for maximum accuracy, resulting in a better error percentage. It is worth noting that the SM-PSO method achieves the highest accuracy among these methods. This is because it utilizes the time-consuming SPICE transistor-level model and MC simulations, which provide more accurate results but require significant computational resources.

Table 6 demonstrates that the proposed DNN-based yield maximization method has the advantage over the PA yield method. Because, the DNN-based method eliminates the need for the linearization step (equations (4) and (5)), resulting in significantly fewer iterations and less total time required for optimization while maintaining higher accuracy. Also, by utilizing a fast mathematical closed-form of circuit performance metrics and the yield function in a gradient-based optimizer, the DNN-based method achieves fewer iterations and reduced total optimization time compared to the free gradient SM-PSO method. The SM-PSO method, which involves running time-consuming transistor-level models existing in the Spice simulator and numerous MC simulations, requires more computational resources and time. Furthermore, the DNN-based yield maximization method outperforms the SM-PSO-PR method. While SM-PSO-PR employs an online training surrogate model in certain iterations to improve speed compared to SM-PSO, it still relies on a free gradient algorithm and online training. As a result, it requires a greater number of iterations and more total time to converge to a satisfactory solution compared to the DNN-based method. The proposed method employs a precise DNN-based model of performance metrics along with a closed-form yield mathematical formula in powerful gradient-based optimizers. This allows the method to converge faster to the optimal point and be more efficient than BO's method, which relies on the acquisition function for optimization. Not using a gradient-based optimizer leads to more iterations and cannot guarantee global convergence. The PA yield method has fewer execution times compared to the SM-PSO, SM-PSO-PR, and ABO methods because it employs a mathematical formula of yield that is independent of time-consuming MC simulations. Additionally, the PA yield method, which utilizes derivative-based optimizers, achieves better solutions with fewer iterations compared to the SM-PSO and SM-PSO-PR methods. It's worth mentioning that the PA yield method incorporates a simulator in some iterations, which contributes to its good accuracy compared to the SM-PSO-PR method and provides an accuracy close to that of the DNN-based yield maximization and SM-PSO methods.

TABLE 7. Real swiping ranges of the optimization constraints (RHS) before normalization for PHFF example.

Performance metric as constraint	RHS swiping range
DQ delay (s)	$[81e-12\ ,\ 84e-12]$
CQ delay (s)	$[55e-12\ ,\ 557e-12]$
Setup time (s)	$[25e-12\ ,\ 33e-12]$
Hold time (s)	$[16e-12\ ,\ 18e-12]$
Dynamic power (w)	$[3.7e-6\ ,\ 4e-6]$

TABLE 8. Upper and lower bounds of the LLTFF design variables.

Variable	P1	P3	P4	N2	N3	N4
UBs (m)	$38e-9$	$97e-9$	$69e-9$	$86e-9$	$38e-9$	$57e-9$
LBs (m)	$28e-9$	$70e-9$	$51e-9$	$64e-9$	$28e-9$	$43e-6$

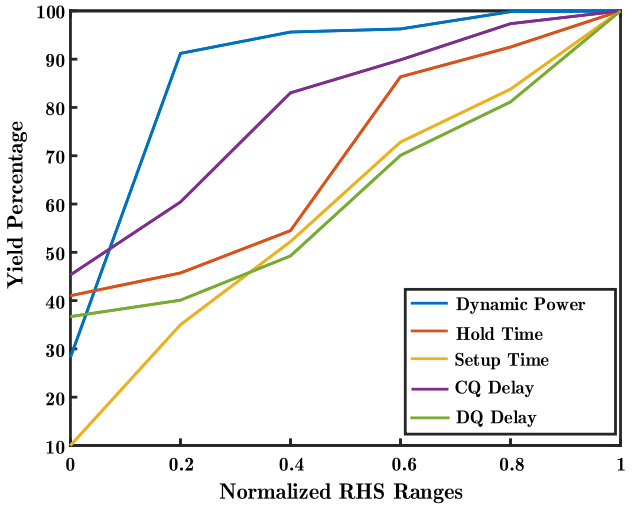


FIGURE 10. Effect of tightening (reducing RHS of the) constraints on yield for PHFF example.

The ABO method demonstrated higher accuracy than the SM-PSO-PR method by retraining the Gaussian process regression model within the feasible region. Its use of BO's efficient acquisition function also resulted in fewer iterations and less time to obtain better solutions compared to the evolutionary SM-PSO and SM-PSO-PR methods. The ABO method approximates the original model, leading to superior accuracy and solutions compared to the PA method. However, the ABO method requires significantly more iterations and optimization time due to the use of non-gradient-based techniques and repeated training of the costly Gaussian process regression model.

Table 6 indicates that the SM-PSO-PR method achieves convergence to the optimal point approximately 15 times faster than the SM-PSO method. This speed advantage is because the SM-PSO-PR method utilizes a surrogate model in some iterations to speed up the optimization process. In contrast, the SM-PSO method relies on a time-consuming simulator model in each iteration. However, it's important to note that the proposed DNN-based technique outperforms both the SM-PSO and SM-PSO-PR methods in terms of speed. The DNN-based yield maximization method exhibits a significant speed advantage of approximately 3.6K times compared to the SM-PSO method, 240 times compared to the SM-PSO-PR method, 16 times compared to the PA yield method, and 111 times compared to the ABO yield method. This speed advantage is attributed to the utilization of a strong gradient-based optimizer in the DNN-based yield maximization method, which significantly reduces the number of iterations required to reach the optimal solution.

Fig 10 illustrates the effect of tightening the PHFF performance metric constraints on the DNN-based yield maximization process. In Fig. 10, constraints of the (19) sweep towards the lowest value obtained during data generation according to Table 7. The ranges of constraints in Table 7 were normalized from zero to one to show them together on a single figure (Fig. 10). Each performance metric constraint in Table 7 was separately tightened in the DNN-based yield maximization process with five steps to observe its effect on the yield value. Tightening the dynamic power in the first four steps has the least effect on manufacturing yield. However, in the fifth step, tightening the dynamic power has a significant impact on the manufacturing yield. Conversely, the CQ delay has the least effect on manufacturing yield in the fifth step. Hold time and DQ delay exhibit similar behavior and significantly impact manufacturing yield in the first three steps. In subsequent steps, only the hold time has the most significant impact.

B. LLTFF EXAMPLE RESULT

Table 8 provides the upper and lower bounds of the design variables and the upper bounds of the LLTFF performance metrics. The average value of the generated data of the performance metrics, as mentioned in equation (20), was used to determine the upper bounds of statistical design. Table 9 presents the nominal point of the LLTFF, including the design variables and performance metrics, along with the obtained yield, which was initially around 84.11%. This yield can be further optimized.

$$\begin{cases} DQ_{DNN} + 3(\sigma_{DQ_{DNN}}) \leq 588e-12\ (s), \\ CQ_{DNN} + 3(\sigma_{CQ_{DNN}}) \leq 338e-12\ (s), \\ ST_{DNN} + 3(\sigma_{ST_{DNN}}) \leq 252e-12\ (s), \\ HT_{DNN} + 3(\sigma_{HT_{DNN}}) \leq 138e-12\ (s), \\ DP_{DNN} + 3(\sigma_{DP_{DNN}}) \leq 0.223e-6\ (w). \end{cases} \tag{20}$$

Moving on to Table 10, it shows the final values of yield, design variables, and relevant performance metrics for all four methods. Again, the DNN-based yield maximization method can converge to better design variables compared to other methods. However, it results in higher values for dynamic power while achieving lower values for other performance metrics. Table 11 reports the error percentage obtained from the verification process when replacing the design variables in the simulator software and running the MC.

TABLE 9. Obtained nominal point from Equation (16) for LLTFF example.

Optimum design variables	Performance metric array					Yield
$[P1, P3, P4, N2, N3, N4](m)$	$DQ(s)$	$CQ(s)$	$ST(s)$	$HT(s)$	$DP(w)$	
$[34, 70, 51, 64, 38, 55]e-09$	$540.17e-12$	$299.40e-12$	$241.88e-12$	$125.27e-12$	$0.219e-6$	84.11

TABLE 10. Comparison of the yield percentage obtained by the proposed DNN-Based, the PA Yield, the SM-PSO, and the SM-PSO-PR yield maximization methods for a LLTFF Example.

Yield maximization methods	Optimum design variables	Optimum performance metric array					Yield
	$[P1, P3, P4, N2, N3, N4](m)$	$DQ(s)$	$CQ(s)$	$ST(s)$	$HT(s)$	$DP(w)$	
SM-PSO	$[32, 80, 59, 64, 37, 54]e-09$	$546.24e-12$	$308.54e-12$	$238.42e-12$	$128.28e-12$	$0.221e-06$	95.92
SM-PSO-PR [25]	$[34, 80, 55, 68, 36, 56]e-09$	$549.94e-12$	$306.71e-12$	$242.55e-12$	$127.43e-12$	$0.220e-06$	98.61
ABO [26]	$[33, 90, 52, 77, 36, 54]e-09$	$551.28e-12$	$303.20e-12$	$238.65e-12$	$133.10e-12$	$0.221e-06$	99.46
PA yield	$[32, 77, 52, 70, 38, 55]e-09$	$534.12e-12$	$298.76e-12$	$236.12e-12$	$124.14e-12$	$0.221e-06$	98.38
Proposed DNN-based yield	$[35, 90, 55, 74, 39, 56]e-09$	$539.40e-12$	$296.54e-12$	$240.64e-12$	$122.51e-12$	$0.222e-06$	100

TABLE 11. Comparison of the Iterations, CPU Time, Speedup, and Accuracy of the Proposed DNN-Based, the PA Yield, the SM-PSO, and the SM-PSO-PR Yield Maximization Methods for a LLTFF Example.

Yield maximization methods	Number of iterations	Total time (min)	Error percentage	Yield based on Table 10	Yield based on MC	Speed up
SM-PSO	340	1984	–	95.92	95.92	1
SM-PSO-PR [25]	286	45	3.83	98.61	94.97	44
ABO [26]	75	19	0.45	99.46	99.01	104
PA yield	61	6	8.77	98.38	90.44	330
Proposed DNN-based yield	28	1.8	0.03	100	99.97	1102

TABLE 12. Real swiping ranges of the optimization constraints (RHS) before normalization for LLTFF example.

Performance metric as constraint	RHS swiping range
DQ delay (s)	$[530.86e-12\ ,\ 588e-12]$
CQ delay (s)	$[296.68e-12\ ,\ 338e-12]$
Setup time (s)	$[233.56e-12\ ,\ 252e-12]$
Hold time (s)	$[120.33e-12\ ,\ 138e-12]$
Dynamic power (s)	$[0.219e-6\ ,\ 0.223e-6]$

The DNN-based yield maximization method demonstrates a lower error percentage compared to the SM-PSO-PR and the ABO yield approaches and achieves a close error rate to the SM-PSO method. This is primarily due to using accurate DNN-based models of constraints in the DNN-based method.

Table 11 provides information on the number of iterations and the total time required for different methods. The proposed DNN-based yield method requires fewer iterations and less total time compared to the PA yield method, mainly due to the elimination of the polyhedral approximation stage. The DNN-based yield maximization method has a distinct advantage over the SM-PSO-PR and SM-PSO methods in terms of iteration efficiency. By leveraging a gradient-based optimizer, the DNN-based method required significantly fewer iterations to achieve optimal results. It needs at least three times fewer iterations than the SM-PSO-PR, SM-PSO, and ABO methods. Furthermore, the SM-PSO-PR method incorporates online training for surrogate models, which enables it to converge to a favorable solution approximately 44 times faster than the SM-PSO method. The ranges of constraints in Table 12 were normalized from zero to one to show them together on a single figure (Fig. 11). In Fig. 11, each performance metric constraint is moved to its lowest value in separate optimization with five steps to demonstrate its impact on the yield value. Hold time and CQ delay have the greatest and least effect on LLTFF yield during this five-step tightening. Dynamic power and CQ delay exhibit the lowest yield loss slope in the LLTFF example.

IV. DISCUSSION

The traditional Monte Carlo method requires running the entire circuit multiple times within each optimization iteration, which is time-consuming and inefficient, especially for more complex circuits. In contrast, DNNs can accurately model complex input-output relationships in circuits even by using real-world measurement data, making them applicable to real-world optimization problems. This capability was supported by the universal approximation theorem. Additionally, error mitigation strategies, such as Dropout, further reduce errors in optimization problems. An optimal structure selection strategy can also be employed to further mitigate errors.

A. DATA GENERATION AND SIMULATIONS

We generate training data using the Spice simulator by varying the desired design variables, training the model, and evaluating the test error initially. If the desired test error is not achieved, we increase the amount of training data. The required number of training data points is determined experimentally.

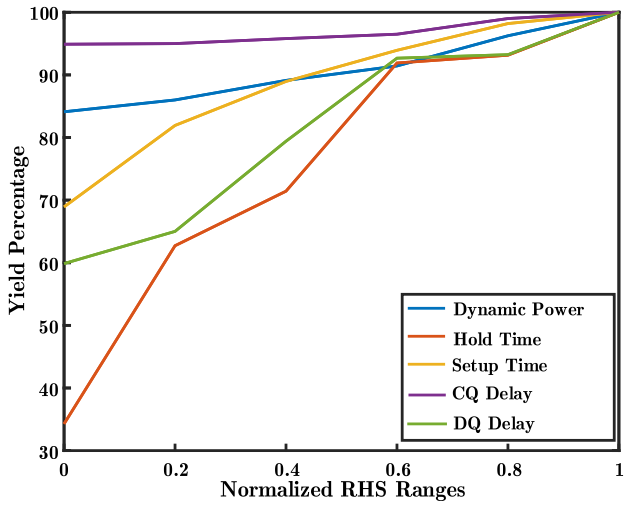


FIGURE 11. Effect of tightening (reducing RHS of the) constraints on yield for LLTFF example.

B. NEURAL NETWORK DEVELOPMENT, FRAMEWORK AND SOFTWARE:

We initiated the process with a compact shallow neural network and progressively increased the neuron count in each layer until reaching a specific threshold. If the network failed to achieve the targeted training error, we added a layer and repeated the process until the network attained the desired accuracy level. The Python programming language, the Keras library, and the Jupyter Notebook framework were used to create, train, and develop the neural network model. A Git repository is available to validate the proposed method, which includes Spice syntax-based code of LLTFF and PHFF examples, neural network training, and yield maximization code. The repository for the proposed yield maximization can be accessed:
https://github.com/Alirezasajjadii/DNN_based_Yield

V. CONCLUSION

In this paper, we addressed the limitations of deterministic circuit design techniques in the presence of manufacturing process variations and time-consuming simulation-based yield maximization methods for statistical design. Traditional deterministic approaches fail to account for process variations, while simulation-based yield maximization methods often suffer from computational inefficiency. To address these issues, we initially explored polyhedral approximation methods which mathematically formulate yield maximization process while linearizing the nonlinear constraints. The PA yield method, utilizing derivative-based optimizers, demonstrated superior solutions with fewer iterations compared to the SM-PSO and SM-PSO-PR, ABO yield maximization methods. However, recognizing the need for further improvement, we introduce a novel DNN-based approach. The proposed DNN-based yield maximization method offers significant advantages over conventional methods such as simulation-based techniques. Compared to the traditional simulation-based methods, the DNN-based approach achieves much faster optimization. In addition to its speedup advantage, the proposed method maintains a high level of accuracy superior to the simulation-based methods while requiring much fewer iterations leading to better manufacturing yield. On top of the above advantages, employing deep neural networks makes us capable of mathematically formulate the original nonlinear constraints without linear/polyhedral approximation leading to further accuracy of the yield optimization process. The scalability of the proposed method is another notable strength. By leveraging the DNNs to model performance metrics, the method can handle complex circuit designs with multiple variables. This scalability makes the proposed method a suitable option for the yield maximization process, even in scenarios involving large-scale circuit designs. The proposed method offers significant advantages in speed, accuracy, convergence, and scalability. These features make it a promising option for optimizing circuit manufacturing yield in various design scenarios.

REFERENCES

[1] S. M. Ebrahimipour, B. Ghavami, and M. Raji, "A statistical gate sizing method for timing yield and lifetime reliability optimization of integrated circuits," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 2, pp. 759–773, Apr. 2021.

[2] X. Wang, C. Yan, Y. Ma, B. Yu, F. Yang, D. Zhou, and X. Zeng, "Analog circuit yield optimization via freeze–thaw Bayesian optimization technique," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 11, pp. 4887–4900, Nov. 2022.

[3] B. Ghavami, M. Raji, M. Ibrami, and L. Shannon, "Reliable circuit design using a fast incremental-based gate sizing under process variation," *IEEE Trans. Device Mater. Rel.*, vol. 22, no. 3, pp. 371–380, Sep. 2022.

[4] H. Zheng and L. Zhou, "A yield prediction method based on nonparametric statistical and comparison with a networks method," *IEEE Trans. Semicond. Manuf.*, vol. 36, no. 1, pp. 144–146, Feb. 2023.

[5] L. Van Brandt, R. Saeidi, D. Bol, and D. Flandre, "Accurate and insightful closed-form prediction of subthreshold SRAM hold failure rate," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 7, pp. 2767–2780, Jul. 2022.

[6] I. Seghaier, M. H. Zaki, and S. Tahar, "Mating sensitivity analysis and statistical verification for efficient yield estimation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 2, pp. 294–307, Feb. 2020.

[7] M. S. Golanbari, S. Kiamehr, M. Ebrahimi, and M. B. Tahoori, "Selective flip-flop optimization for reliable digital circuit design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 7, pp. 1484–1497, Jul. 2020.

[8] Y.-L. Chen, W.-R. Wu, C. J. Liu, and J. C. Li, "Simultaneous optimization of analog circuits with reliability and variability for applications on flexible electronics," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 1, pp. 24–35, Jan. 2014.

[9] J.-F. Lin, Z.-J. Hong, J.-T. Wu, X.-Y. Tung, C.-H. Yang, and Y.-C. Yen, "Low-voltage and low-power true-single-phase 16-transistor flip-flop design," *Sensors*, vol. 22, no. 15, p. 5696, Jul. 2022.

[10] F. Centurelli, G. Scotti, A. Trifiletti, and G. Palumbo, "Design of low-voltage power efficient frequency dividers in folded MOS current mode logic," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 680–691, Feb. 2021.

[11] S. A. Sadrossadat, H. Mostafa, and M. Anis, "Statistical design framework of submicron flip-flop circuits considering process variations," *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 1, pp. 69–79, Feb. 2011.

[12] A. K. Mishra, U. Chopra, and D. Vaithiyanathan, "A partially static high frequency 18T hybrid topological flip-flop design for low power application," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1592–1596, Mar. 2022.

[13] Y. Cao. (2012). *Predictive Technology Model*. [Online]. Available: http://ptm.asu.edu/

[14] A. Stefanidis, D. Mangiras, C. Nicopoulos, D. Chinnery, and G. Dimitrakopoulos, "Autonomous application of netlist transformations inside Lagrangian relaxation-based optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 8, pp. 1672–1686, Aug. 2021.

[15] S. Banerjee, A. Chaudhuri, A. Ning, and K. Chakrabarty, "Variation-aware delay fault testing for carbon-nanotube FET circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 2, pp. 409–422, Feb. 2021.

[16] R. Krishnan, W. Wu, S. Bodapati, and L. He, "Accurate multi-segment probability density estimation through moment matching," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, early access, May 4, 2016, doi: 10.1109/TCAD.2016.2562923.

[17] F. Frustaci, P. Corsonello, and S. Perri, "Analytical delay model considering variability effects in subthreshold domain," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 3, pp. 168–172, Mar. 2012.

[18] M. Nabavi and M. Shams, "A gate sizing and transistor fingering strategy for subthreshold CMOS circuits," *IEICE Electron. Exp.*, vol. 9, no. 19, pp. 1550–1555, 2012.

[19] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1533–1545, Oct. 2013.

[20] R. Saeidi, M. Sharifkhani, and K. Hajsadeghi, "Statistical analysis of read static noise margin for near/sub-threshold SRAM cell," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 12, pp. 3386–3393, Dec. 2014.

[21] S. Yin, G. Dai, and W. W. Xing, "High-dimensional yield estimation using shrinkage deep features and maximization of integral entropy reduction," in *Proc. 28th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2023, pp. 283–289.

[22] S. Weber and C. Duarte, "Yield analysis for electrical circuit designs: Many problems and some recent developments in electronic engineering," *IEEE Solid StateCircuits Mag.*, vol. 12, no. 1, pp. 39–52, Winter. 2020.

[23] S. Zhang, F. Yang, D. Zhou, and X. Zeng, "Bayesian methods for the yield optimization of analog and SRAM circuits," in *Proc. 25th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2020, pp. 440–445.

[24] M. Wang, F. Yang, C. Yan, X. Zeng, and X. Hu, "Efficient Bayesian yield optimization approach for analog and SRAM circuits," in *Proc. 54th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2017, pp. 1–6.

[25] U. B. Vyas and V. A. Shah, "Optimisation based 3-dimensional polynomial regression to represent lithium-ion battery's open circuit voltage as function of state of charge and temperature," *J. Energy Storage*, vol. 50, Jun. 2022, Art. no. 104656.

[26] M. Wang, W. Lv, F. Yang, C. Yan, W. Cai, D. Zhou, and X. Zeng, "Efficient yield optimization for analog and SRAM circuits via Gaussian process regression and adaptive yield estimation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 10, pp. 1929–1942, Oct. 2018.

[27] W. Zhao, Y. Ha, and M. Alioto, "Novel self-body-biasing and statistical design for near-threshold circuits with ultra energy-efficient AES as case study," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 8, pp. 1390–1401, Aug. 2015.

[28] Z. Abbas, M. Olivieri, U. Khalid, A. Ripp, and M. Pronath, "Optimal NBTI degradation and PVT variation resistant device sizing in a full adder cell," in *Proc. 4th Int. Conf. Rel., InfoCom Technol. Optim. (ICRITO) (Trends Future Directions)*, Sep. 2015, pp. 1–6.

[29] Z. Heshmatpour, L. Zhang, and H. M. Heys, "Robust CNFET circuit sizing optimization," in *Proc. Int. Symp. VLSI Design, Autom. Test (VLSI-DAT)*, Apr. 2022, pp. 1–4.

[30] K. Agarwal, A. Jain, D. Amuru, and Z. Abbas, "Fast and efficient ResNN and genetic optimization for PVT aware performance enhancement in digital circuits," in *Proc. Int. Symp. VLSI Design, Autom. Test (VLSI-DAT)*, Apr. 2022, pp. 1–4.

[31] W.-E. Wei, H.-Y. Li, C.-Y. Han, J. C. Li, J.-J. Huang, I.-C. Cheng, C.-N. Liu, and Y.-H. Yeh, "A flexible TFT circuit yield optimizer considering process variation, aging, and bending effects," *J. Display Technol.*, vol. 10, no. 12, pp. 1055–1063, Dec. 2014.

[32] A. Shafaei, S. Chen, Y. Wang, and M. Pedram, "A cross-layer framework for designing and optimizing deeply-scaled FinFET-based SRAM cells under process variations," in *Proc. 20th Asia South Pacific Design Autom. Conf.*, Jan. 2015, pp. 75–80.

[33] A. Sanabria-Borbón, S. Soto-Aguilar, J. Estrada-López, D. Allaire, and E. Sánchez-Sinencio, "Gaussian-process-based surrogate for optimization-aided and process-variations-aware analog circuit design," *Electronics*, vol. 9, no. 4, p. 685, Apr. 2020.

[34] A. Canelas, R. Póvoa, R. Martins, N. Lourenço, J. Guilherme, J. P. Carvalho, and N. Horta, "FUZYE: A fuzzy *C*-means analog IC yield optimization using evolutionary-based algorithms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 1, pp. 1–13, Jan. 2020.

[35] W. Hu, Z. Wang, S. Yin, Z. Ye, and Y. Wang, "Sensitivity importance sampling yield analysis and optimization for high sigma failure rate estimation," in *Proc. 58th ACM/IEEE Design Autom. Conf. (DAC)*, Dec. 2021, pp. 895–900.

[36] Y. Liu, G. Dai, Y. Cheng, W. Kang, and W. W. Xing, "OPT: Optimal proposal transfer for efficient yield optimization for analog and SRAM circuits," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Oct./Nov. 2023, pp. 1–9.

[37] Y. Liu, G. Dai, and W. W. Xing, "Seeking the yield barrier: High-dimensional sram evaluation through optimal manifold," in *Proc. 60th ACM/IEEE Design Automat. Conf. (DAC)*, Jul. 2023, pp. 1–6.

[38] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[39] Z. Tibenszky, C. Carta, and F. Ellinger, "A 0.35 mW 70 GHz divide-by-4 TSPC frequency divider on 22 nm FD-SOI CMOS technology," in *Proc. IEEE Radio Freq. Integr. Circuits Symp. (RFIC)*, Aug. 2020, pp. 243–246.

[40] J. Leong, K. Ponnambalam, J. Binns, and A. Elkamel, "Thermally constrained conceptual deep geological repository design under spacing and placing uncertainties," *Appl. Sci.*, vol. 11, no. 24, p. 11874, Dec. 2021.

[41] S. W. Director, P. Feldmann, and K. Krishna, "Optimization of parametric yield: A tutorial," *Int. J. High Speed Electron. Syst.*, vol. 3, no. 1, pp. 95–136, 1992.

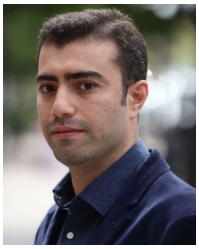
[42] A. Faraji, S. A. Sadrossadat, W. Na, F. Feng, and Q.-J. Zhang, "A new macromodeling method based on deep gated recurrent unit regularized with Gaussian dropout for nonlinear circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 7, pp. 2904–2915, Jul. 2023.

[43] L. Long and X. Zeng, "Beginning deep learning with tensorFlow: Work with keras," in *MNIST Data Sets, and Advanced Neural Networks*. New York, NY, USA: Apress, 2022, pp. 1–978, doi: 10.1007/978-1-4842-7915-1. [Online]. Available: https://www.apress.com/gp

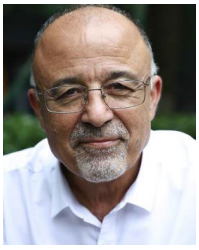
SAYED ALIREZA SAJJADI (Member, IEEE) received the B.Sc. degree from the Azad University of Kazerun, Kazerun, Iran, in 2020, and the master's degree in electrical engineering from the Department of Electrical Engineering, Yazd University, Yazd, Iran. His research interests include computer-aided design, neural networks, digital design, VLSI design and optimization, probabilistic design, yield maximization, and their applications in circuit designs.

SAYED ALIREZA SADROSSADAT (Senior Member, IEEE) received the B.Sc. degree from the University of Tehran, Tehran, Iran, in 2007, the master's degree from the University of Waterloo, Waterloo, ON, Canada, in 2010, and the Ph.D. degree from Carleton University, Ottawa, ON, Canada, in 2015. He is currently the Chair of the Artificial Intelligence Group, Department of Computer Engineering, Yazd University, Yazd, Iran. His current research interests include optimization and neural network-based modeling of linear/nonlinear components and circuits, computer-aided design, deep learning, very large-scale integration design, probabilistic design, and yield maximization. He has been a Technical Reviewer for several IEEE/IET journals, such as IEEE Transactions on Neural Network and Learning Systems, IEEE Transactions on Microwave Theory and Technique, IEEE Transactions on Very Large Scale Integration Systems, *IET Microwave and Propagation*, *IET Electronic Letters*, and IEEE Canadian Journal of Electrical and Computer Engineering.

ALI MOFTAKHARZADEH received the B.Sc. degree in electrical engineering from the K. N. Toosi University of Technology, Tehran, Iran, in 2002, and the M.Sc. and Ph.D. degrees in electrical engineering from the Sharif University of Technology, Tehran, in 2004 and 2009, respectively. In 2010, he joined the Department of Electrical Engineering, Yazd University, Yazd, Iran. His current research interests include digital system design, DSP, image processing, and linear/nonlinear circuits macromodeling.



MORTEZA NABAVI (Member, IEEE) received the B.S. degree in computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2008, the M.A.Sc. degree in computer engineering from Boston University, Boston, MA, USA, the M.A.Sc. degree in electronics from Carleton University, Ottawa, ON, Canada, in 2012, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada. He joined the Research and Development Group, Sidense Company, Ottawa, where he was involved in designing subthreshold one-time programmable (OTP) memories. His current research interests include subthreshold circuits and memories.



MOHAMAD SAWAN (Life Fellow, IEEE) received the Ph.D. degree from the University of Sherbrooke, Canada. He is currently a Chair Professor with Westlake University, Hangzhou, China; and an Emeritus Professor with Polytechnique Montréal, Canada. He is also the Founder and the Director of the Center of Excellence in Biomedical Research on Advances-on-Chips Neurotechnologies (CenBRAIN Neurotech), Westlake University, and the Polystim Neurotech Laboratory, Polytechnique Montréal. He has published more than 1000 peer-reviewed articles, one handbook, three books, 13 book chapters, 12 patents, and 20 other patents are pending. He is a fellow of the Royal Society of Sciences of Canada (FRSC), the Canadian Academy of Engineering, the Engineering Institutes of Canada, and the Asia–Pacific Artificial Intelligence Association (AAIA); and an ''Officer'' of the National Order of Quebec. He is also the Co-Founder, an Associate Editor, and was the Editor-in-Chief of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, from 2016 to 2019. He is also the Founder of the Flagship IEEE International NEWCAS Conference and the Co-Founder of the International IEEE-BioCAS and IEEE-AICAS Conferences. He received several awards, among them the Barbara Turnbull Award from Canadian Institutes of Health Research (CIHR), the J. A. Bombardier and Jacques-Rousseau Awards from the ACFAS, the Queen Elizabeth II Golden Jubilee Medal, the Medal of Merit from the President of Lebanon, Shanghai International Collaboration Award, Chinese Government Friendship Award, and the Hangzhou Outstanding Talent Award. He was the General Chair hosting the 2016 IEEE International Symposium on Circuits and Systems (ISCAS) and the 2020 IEEE International Medicine, Biology and Engineering Conference (EMBC). He was awarded the Canada Research Chair in Smart Medical Devices, from 2001 to 2015 and was leading the Microsystems Strategic Alliance of Quebec (ReSMiQ), Canada, from 1999 to 2018.