Springer Analog Integrated Circuits and Signal Processing Journal manuscript No.
(will be inserted by the editor)

Polynomial Metamodel Based Fast Optimization of Nano-CMOS Oscillator Circuits

Saraju P. Mohanty · **Elias Kougianos**

Received: 11 Jan 2013 / Revised: 03 Dec 2013 / Accepted: 27 Feb 2014

Abstract Modern consumer electronics are designed as Analog/Mixed-Signal Systems-on-Chip (AMS-SoCs). In an AMS-SoC, the analog and mixed-signal portions have not received systematic attention due to their complex nature and the fact that their optimization and simulation consume significant portions of the design cycle time. This paper presents a new approach to reduce the design cycle time by combining accurate polynomial metamodels and optimization algorithms. The approach relies on a mathematical representation (metamodel or surrogate model) of AMS-SoC subsystems/components. Polynomial metamodels are created from post-layout parasitic netlists and provide an accurate representation for each Figure-of-Merit (FoM) over the entire design space of the AMS-SoC component. The metamodel approach saves a very significant amount of time during design iterations. Polynomial metamodels are reusable and language independent. Three algorithms are investigated to compare the speed for optimization on the polynomial metamodels. Two widely used circuits have been designed in two different technologies as comparative case studies: an 180 nm LC-VCO and a 45 nm ring oscillator. Experimental results prove that the metamodel-based optimization achieved speed-up as high as 21,600× for the LC-VCO circuit and 11,750× for the ring oscillator in comparison to the actual circuit netlist-based (SPICE) optimization, with less than 1% error. Thus, the paper demonstrates that the polynomial metamodeling approach to the design problem is an effective and accurate means for fast design space exploration and optimization.

Keywords Mixed-Signal Circuits, Oscillator Circuits, Polynomial Metamodeling, Performance Optimization, Optimization Algorithms

Computer Science and Engineering, University of North Texas, Denton, TX 76203.
Tel.: +1 940-565-3276
Fax: +1 940-565-2799
E-mail: saraju.mohanty@unt.edu

Engineering Technology, University of North Texas, Denton, TX 76203.
Tel.: +1 940-891-6708
Fax: +1 940-565-2666
E-mail: elias.kougianos@unt.edu

1 Introduction and Motivation

Modern consumer electronics (e.g. mobile phones) are built as Analog/Mixed-Signal Systems-On-Chip (AMS-SoCs) in which the analog and digital portions are integrated on the same die for cost-performance tradeoffs [7,20]. Analog components are absolutely necessary in an AMS-SoCs, at a minimum as interface elements, even though a significant amount of computing is being performed by digital subsystems. In addition, components such as Analog to Digital Converters (ADCs), Digital to Analog Converters (DACs), and Phase-Locked Loops (PLLs) are intrinsically mixed-signal circuits. Present day AMS-SoCs are of gigascale complexity and consist of nanoscale CMOS transistors [41]. To make the situation worse for such highly complex systems, the time-to-market has been reduced significantly due to strong competition. In such a situation, Computer Aided Design (CAD) frameworks are more important than ever in order to produce error free AMS-SoCs on time [7,15].

The standard circuit-level simulation based design approach can only be used when a designer has sufficient time for running simulations to optimize the circuit. Unfortunately, this is usually not the case, since the timeline for a design process is very short. Usually simulation times for complex circuits are very long and it is not feasible to conduct an exhaustive search to find the optimal circuit. The *simulation time for PLL lock on a full parasitic netlist is of the order of many days to weeks* [6]. Standard simulation based flows do not account for the parasitics of the circuit, which can only be calculated after the initial physical design is complete. The parasitics present in the circuit have a dramatic effect on the responses of the circuit [32]. Analyzing the usual design process, one can see that the longest manual labor is spent during the physical layout creation and numerous subsequent adjustments. Therefore, to shorten the design time, this paper introduces a design flow that requires only two iterations of the physical layout. The first is the starting physical design and the second is the final design after the optimization phase.

The quest for accurate and less time consuming design flows has led to the metamodeling approach which is widely used in other technical fields [25,38,10]. The objective of this paper is to simplify the optimization phase of the design cycle and perform it using the least possible amount of circuit simulations possible. Optimization over metamodels is proposed as a solution for fast and yet accurate nano-CMOS circuit design exploration. Metamodels are essentially an abstract representation, e.g. commonly a mathematical algorithm, of the behavior of the circuit for a desired output [43]. It may be noted that in the existing literature the *terms macromodel and metamodel are often used interchangeably but they represent very distinct approaches* [1,30]. Macromodels are reduced complexity models of the circuit but rely on the same type of modeling and simulator as the original model (e.g., SPICE) [39]. In the metamodeling approach, the underlying system is completely decoupled from the simulator and the resulting metamodel (i.e., model of a model) is a mathematical algorithm, as depicted in Fig. 1 [12]. A metamodel is more general, flexible and easier to simulate and optimize than a macromodel.

The metamodels available in the existing literature are of many types. A selected taxonomy of common metamodels is presented in Fig. 2 [47,40,27]. The different metamodel classes include polynomials, Splines, artificial neural networks (ANN), support vector machines (SVM), genetic programming, Kriging methods, and Gaussian processes. The choice of a specific metamodel is a tradeoff between speed and accuracy. Polynomial metamodels are easy to create, but have lower accuracy for complex circuits and large number of design parameters. Nonpolynomial metamodels are difficult to create but can handle complex circuit with large number of design parameters. An accurate metamodel simplifies the calculations for the effects of the circuit and makes it possible to employ more time consuming

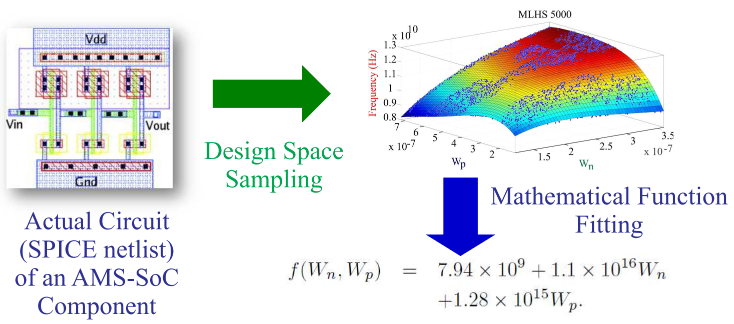

Vdd
Vin
Vout
Gnd
Actual Circuit (SPICE netlist) of an AMS-SoC Component
Design Space Sampling
MLHS 5000
Frequency (Hz)
x 10^10
1.3
1.2
1.1
1
0.9
0.8
Wp
x 10^-7
Wn
x 10^-7
Mathematical Function Fitting
$f(W_n, W_p) = 7.94 \times 10^9 + 1.1 \times 10^{16} W_n + 1.28 \times 10^{15} W_p.$

Fig. 1 Metamodeling transforms an AMS-SoC component netlist to mathematical functions thus facilitating fast design exploration.

accurate optimization algorithms. A comparative perspective of polynomial metamodeling and an ANN for a 180 nm CMOS phase-locked loop (PLL) is presented in Table 1 [29,30]. Polynomial metamodels with approximately 1% loss of accuracy provide simpler mathematical forms to handle reasonable size circuits. This paper shows that polynomial metamodels can be used to adjust the initial physical design circuit to the target figure of merit. Then, selected optimization algorithms are explored to search the design space for a target design specification. To account for parasitics, the physical designs are initially created and a netlist with full parasitics is generated for subsequent analysis.

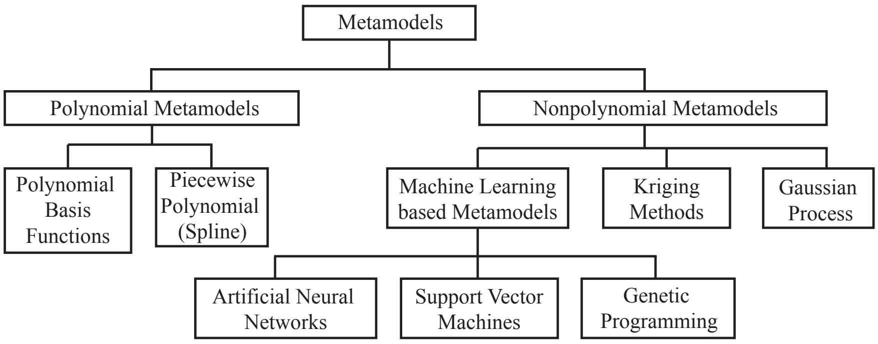


Fig. 2 Different Types of metamodels which can be explored for VLSI design.

Table 1 Comparative Perspective of Polynomial and ANN Metamodels for a 180 nm CMOS PLL with center frequency f_c of 2.7GHz. RMSE is the root mean square error of the approximation considering SPICE simulation as the “accurate” result.

Polynomial Metamodel		ANN Metamodel	
Time to Create	RMSE	Time to Create	RMSE
11 hr for sampling + 1 min for creation.	78 MHz (2.9% of f_c)	11 hr for sampling + 10 min for creation.	48 MHz (1.8% of f_c)

The rest of the paper is organized as follows. The novel contributions of this paper are listed in Section 2. Section 3 presents related research. The case study nanoscale circuits are presented in Section 4. Section 5 presents the metamodel-based design flow. Section 6 presents the three algorithms investigated in the metamodel-based design flow. Experimental results are discussed in Section 7. Section 8 concludes this paper and discusses future research.

2 Contributions of this Paper

This paper introduces an approach called polynomial metamodel-based optimization to acquire an accurate physical design (layout) of AMS-SoC subsystems with minimal design cycle time. A polynomial metamodel is a mathematical description of a figure of merit of a circuit in terms of it design parameters. The creation of an accurate metamodel provides designers with a simple, less computationally expensive, reusable and language-independent model which is sufficiently accurate to produce an optimized result for the given parametric problem. The approach is depicted in Fig. 3.

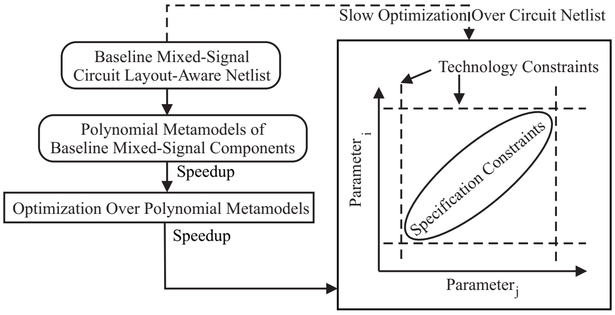

Slow Optimization Over Circuit Netlist
Baseline Mixed-Signal Circuit Layout-Aware Netlist
Polynomial Metamodels of Baseline Mixed-Signal Components
Speedup
Optimization Over Polynomial Metamodels
Speedup
Technology Constraints
Specification Constraints
Parameter i
Parameter j

Fig. 3 Fast design space exploration of analog circuits through accurate metamodeling. *The exploration and comparison of optimization algorithms over the metamodels is the scope of this paper.*

For maximum accuracy and optimal design, one can optimize the actual circuit model (a SPICE netlist). This optimization on the actual circuit, represented as a dashed line in Fig. 3, is very slow and may be even impossible for complex nanoscale circuits with large numbers of transistors and extensive interconnects. For fast, yet accurate design optimization of analog circuits the proposed approach is demonstrated by the solid line in Fig. 3. In this approach, polynomial metamodels of the AMS-SoC components are first generated. The component optimization is then performed on the polynomial metamodels instead of the actual circuit (i.e. parasitic aware netlist). This makes the design exploration fast and yet accurate as the polynomial metamodels are *ab initio* generated accounting for parasitic effects. The current paper focuses on exploration of the optimization algorithms on metamodels, whereas the details of metamodel generation are presented in our previous publication [12]. Thus, *the current paper and [12] jointly cover the complete spectrum of simulation and optimization of the overall design cycle*.

To the best of the authors' knowledge, this is *the first application of polynomial metamodels to parasitic-aware netlists for analog/mixed-signal design optimization*. The *novel contributions of this paper* are summarized as follows:

1. This paper proposes a metamodel-based design flow for fast and accurate optimization of nano-CMOS complex mixed-signal circuits.
2. The design flow requires a single iteration with two manual layouts, the second of which is only a perturbation of the first; thus the proposed flow reduces errors related to layout iterations and shortens the design cycle.
3. As a step toward optimization of nano-CMOS technology based circuits, four distinct optimization algorithms are discussed which are based on the following: exhaustive search, simulated annealing, tabu search, and artificial bee colony (ABC).
4. An 180 nm LC-VCO and a 45 nm Ring Oscillator (RO) are designed and characterized, including the layout, and are used as case studies.
5. Full RCLK (resistance, capacitance, and self and mutual inductance) parasitic extraction is performed and compared to the schematic of the oscillators. Metamodels are generated on the parasitic extracted netlist from the initial physical layout.
6. It is shown that metamodeling is significantly faster compared to SPICE simulations.
7. The use of optimization techniques with and without metamodeling are compared.
8. The metamodeling approach is used to center the LC-VCO and ring oscillator to various target frequencies.

3 Related Prior Research

Research is in full swing to reduce design cycle time of analog/mixed-signal circuits and systems. This section discusses research works that relate to the scope of the present paper. They are grouped into three categories: (1) optimization on actual circuits (i.e. netlists), (2) design effort reduction using macromodels, and (3) design time reduction using metamodels.

Optimization on actual circuits (netlists) has the advantage of using the same framework for design, simulation, and characterization. However, they are often not scalable to large circuits. In [8], automatic synthesis of CMOS LDO regulators is presented. In [14], a conjugate gradient optimization is presented for current-starved VCO circuits. In [5], a geometric program formulation for transistor sizing and optimization is investigated but it is not parasitic aware and slow to converge. In [44], a framework based on ellipsoidal uncertainty is presented. Design characterization of ring oscillators including jitter, power and frequency is presented in [33]. In [46], geometric programming is used for metal mask configurable circuit optimization. A heuristic tabu search optimization algorithm is proposed in [2] and compared with simulated annealing for an operational amplifier. In [26], an automated analog circuit design was proposed for a two-stage CMOS operational amplifier.

Macromodels have been extensively explored as an attractive approach in analog design. In [4], a variation-aware performance macromodeling technique is presented for analog building blocks to facilitate synthesis. In [37], an approach for passivity verification and enforcement of large order macromodels is presented for scattering parameters based on multiport subnetworks. In [18], the limitations of macromodels that can lead to unacceptably high estimation errors are discussed. In [9], a sequential design space decomposition technique is presented for efficient analog feasibility and performance macromodeling. In [45], a leakage current macromodeling technique for dual-threshold circuits is presented.

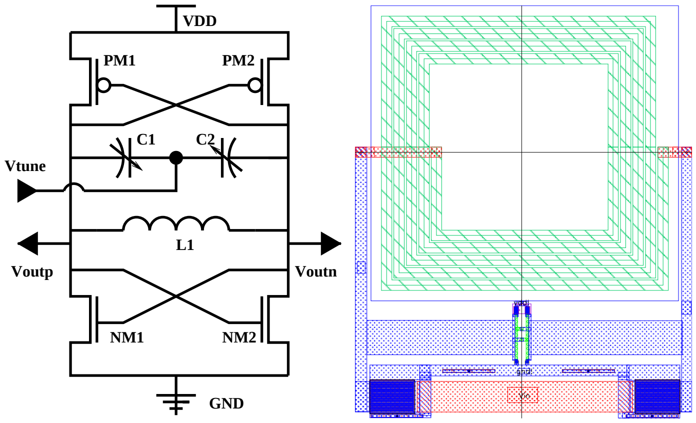
A few existing research works deal specifically with metamodeling, but not necessarily for mixed-signal circuits. Adaptive or sequential metamodeling has been implemented for grid computing in [17]. A surrogate modeling approach is also used for statistical wirelength estimation in [42]. An automated technique for surrogate multivariate mathematical modeling for microwave components is proposed and tested in [21]. In [24], artificial neural networks have been used to model spiral inductors and particle swarm optimization (PSO) has been used for optimization. IP reuse for SoC interaction and microprocessor design is covered in [25]. In [34] surrogate modeling for expensive circuit-level simulation is proposed for support vector machine (SVM)-based machine learning. In [23], ANN and Gaussian processes has been used for performance prediction without using electromagnetic (EM) simulations. In [22], Gaussian process has been used along with memetic optimization for differential amplifier synthesis.

4 The Case Study Nano-CMOS Circuits: LC-VCO and Ring Oscillator

This section presents two case study oscillator circuits, an LC-VCO and a ring oscillator. Both of these are widely used in mixed-signal circuits and systems. These are designed for two different technology nodes. Complete design details are given in [12], however they are briefly discussed here again for completeness of the current paper.

4.1 A 180 nm CMOS LC-VCO Circuit

A conventional complementary NMOS and PMOS cross-coupled LC-VCO circuit (Fig. 4(a)) is used as case study. The inductor is chosen to be as large as possible to minimize power consumption, which leaves the varactor capacitance and the transistor sizing as design variables. It is logical to find a close value for the varactors and the inductor first then adjust the width of the transistors to "fine-tune" the needed frequency as changing the size of transistors does not change their capacitance significantly relative to the varactors.



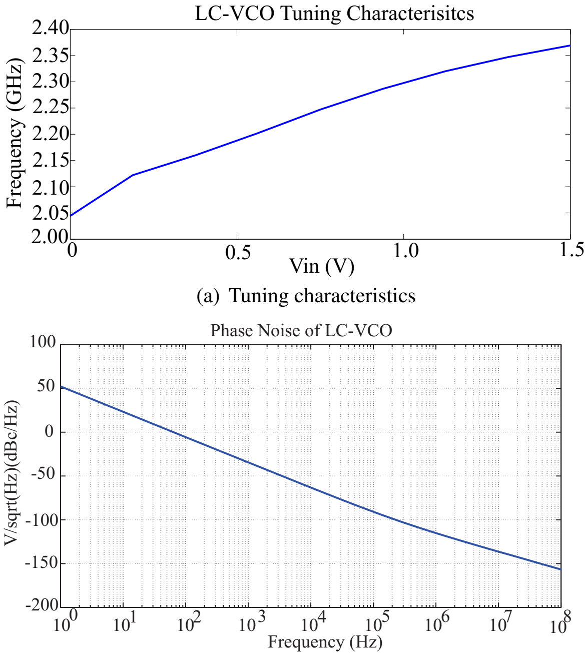
(a) Schematic design

(b) Layout design

Fig. 4 Design of the LC-VCO for 180 nm CMOS technology.

The physical design of the LC-VCO is shown in Fig. 4(b). The parasitic-aware netlist of the LC-VCO is extracted from the physical design which is made to be symmetrical. The symmetry of the layout provides even-order distortion in the differential output waveform and up-conversion [16]. The wire widths are maximized to minimize wire resistance and provide some space for future sizing of the devices as the layout will be adjusted in the second iteration of the design optimization flow.

The tuning characteristics of the designed LC-VCO, which has a target frequency of 2.2 GHz in the middle of the tuning range, are shown in Fig. 5(a). The phase noise of the LC-VCO is shown in Fig. 5(b) for the center frequency of 2.2 GHz. Phase noise at 1 MHz offset from the carrier is -117 dBc/Hz. This circuit is completely symmetrical for V_{outn} and V_{outp} to have exactly the same frequency. The transistors have the same effect on frequency as the varactors.



(a) Tuning characteristics

(b) Phase noise of LC-VCO at 2.2 GHz.

Fig. 5 Characterization of the 180 nm LC-VCO.

The simulation on the parasitic extracted netlist with the same sizing of the devices as in the schematic demonstrates the effect of parasitics on the circuit. The number of extra components which have been added due to the parasitics is presented in Table 2. An implicit assumption is that minor modifications on some devices in this particular layout will not have drastic effects on the parasitics of the future modified layout. Hence, the optimization can be performed on the sizing of the devices of this circuit with extracted parasitics and only consider the parasitics in the final layout to assess the effects they have on the FoMs during the optimization.

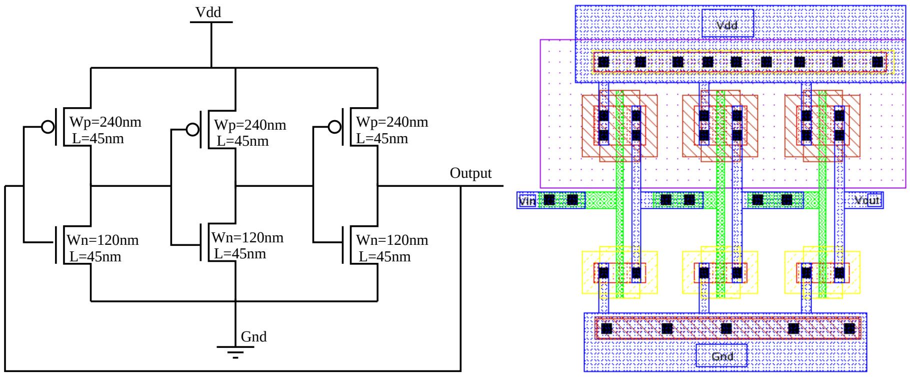
4.2 A 45 nm CMOS Ring Oscillator Circuit

A Ring Oscillator (RO) consists of an odd number of inverters connected in series with positive feedback as shown in Fig. 6(a). In this circuit, the transistor widths are varied to

Table 2 Number of parasitic elements of the 180 nm LC-VCO.

Simulations	Transistors	Capacitors	Inductors	Resistors	Total
Without Parasitics	2	2	1	0	5
With Parasitics	2	108	14	600	724

obtain the desired oscillating frequency. The initial design parameters are as follows: length of transistors L = 45 nm, width of NMOS W_n = 120 nm and width of PMOS W_p = 240 nm, with a nominal operating voltage of V_{dd} = 1 V, as shown in Fig. 6(a). It is difficult to estimate the effect of parasitics without performing the layout even for such a simple circuit. A full parasitic netlist is extracted from the physical design of the ring oscillator, which is shown in Fig. 6(b). The extracted netlist is then parameterized and is used subsequently for all parasitic-aware design, simulations and optimization.



(a) Schematic design (b) Layout design

Fig. 6 Design of the ring oscillator for 45 nm CMOS technology.

SPICE simulation of the post-layout parasitic netlist of the ring oscillator circuit shows a dramatic decrease in frequency compared to the initial schematic design. The presence of parasitics in the circuit also increases the run time of each simulation. In more complex circuits with hundreds or even thousands of transistors, the simulation time will be in the order of days if not weeks, depending on the complexity of the circuit. Table 3 shows a comparison of the number of components between the regular schematic and the parasitic netlist. As evident from the table, the complexity of the circuit (SPICE topology matrix) has increased by almost 20×.

Table 3 Number of parasitic elements of the 45 nm ring oscillator.

Simulation	Transistors	Capacitors	Resistors	Total
Without Parasitics	6	0	0	6
With Parasitics	6	82	19	107

5 Polynomial Metamodel-Based Design Flow using Only Single Layout-Iteration

5.1 Parasitic-Aware Design Optimization Flow

A parasitic-aware design flow that uses *only two manual layouts* is proposed in Fig. 7. An initial physical design is generated once the logical design is complete and meets the specifications. The physical design is then subjected to Design Rule Check (DRC), Layout versus Schematic (LVS), and parasitic (RLCK) extraction. If the specifications are not met, a parasitic parameterized netlist is then created with the design variables as parameters. *The parasitic-aware netlist contains all the parasitics associated with the initial physical design, while the devices that are selected to be varied are parameterized.* The parasitics are captured based on the process design kit specific to a technology. In the current paper the parasitic-aware netlist contains parasitic resistance, capacitance, and self and mutual inductance, along with the active devices. The netlist is simulated using SPICE. The speed up in the design flow stems from the following: (1) use of more automatic iterations instead of manual iterations in the flow and (2) use of fast optimization algorithms.

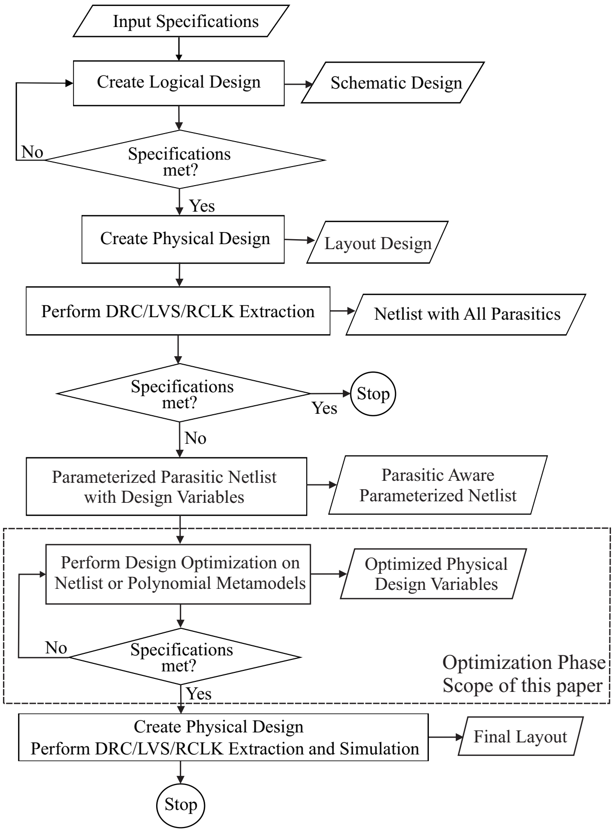

Input Specifications
Create Logical Design
Schematic Design
No
Specifications met?
Yes
Create Physical Design
Layout Design
Perform DRC/LVS/RCLK Extraction
Netlist with All Parasitics
Specifications met?
Yes
Stop
No
Parameterized Parasitic Netlist with Design Variables
Parasitic Aware Parameterized Netlist
Perform Design Optimization on Netlist or Polynomial Metamodels
Optimized Physical Design Variables
No
Specifications met?
Optimization Phase Scope of this paper
Yes
Create Physical Design Perform DRC/LVS/RCLK Extraction and Simulation
Final Layout
Stop

Fig. 7 Proposed parasitic-aware physical design optimization flow using polynomial metamodels. This flow performs the mixed-signal design optimization in one additional layout iteration, in which only two manual layouts are needed instead of multiple iterations.

The resulting parasitic-aware netlist is then used by an automated process to conduct the optimization phase which determines the variable values for the parameters that were chosen for each circuit. The optimization phase is outlined by the dashed line in Fig. 7 and is discussed in this paper. The final physical design is then created by using the optimized parameters. The physical design creation, which is a very time consuming process, needs to be only conducted at most two times using this proposed approach. The perturbation introduced to the layout by the optimization phase does not have a large impact on the parasitic behavior. This approach is justified by the results presented in the following sections.

One of the most important phases of the design flow is the optimization algorithm. The optimization over actual circuits, as is presented in [14], is possible but is still time consuming for large circuits. The current paper will thus optimize the mixed-signal components over their polynomial metamodels.

At this phase of the design flow, the parasitic circuit's netlist is modified to include variable parameters that affect the circuit's output the most. For the LC-VCO, for easier final physical design, the inductor and the varactors are the biggest devices in the circuit and are kept constant because their change will affect the parasitics of the final layout substantially. The varying parameters for optimization are chosen to be W_n and W_p, the NMOS and PMOS widths, respectively. To simplify the study of the algorithms the current paper uses W_n and W_p as the design variables for both circuits. However, the algorithms and the design flow are generic in nature and can accommodate any other design variable that the design engineer intends to use.

5.2 Polynomial-Metamodel Based Design Optimization Flow

The polynomial metamodel based fast and accurate design flow is presented in Fig. 8. These proposed steps are the essence of the creation of an accurate metamodel and the optimization on that metamodel to find the desired response. The parasitic aware design flow is modified to include the details of the polynomial metamodel generation.

An accurate metamodel provides designers with a good understanding of the behavior of mixed-signal components as their design spaces are traversed. The metamodels chosen for this comparison follow a polynomial functional form as follows: $y(x_1,x_2) = \sum_{i,j=0}^{k}(a_{ij}x_1^i x_2^j)$, where y is the response or FoM. $x = [x_1,x_2] = [W_n,W_p]$ is the vector of design variables. a_{ij} are the coefficients determined by the polynomial regression which is chosen to have at most degree $k = 4$. The response FoM is frequency f, power P, or power over frequency ratio (PFR).

Once the metamodels are generated, the designer can conduct more extensive optimization of the circuit and use the same metamodel for different criteria to be optimized. As can be seen from the example in Section 4, the simulation is significantly more time consuming than using the metamodel. The designer can adjust and change the optimization algorithm to fit the proposed design flow, especially if the circuit undergoes multiple optimization (automatic) iterations. Metamodeling is also useful to determine which optimization algorithm to select and adjust for a given circuit. Metamodel sampling is a time consuming process, but by choosing the right sampling and fitting technique, it is possible to apply computationally expensive algorithms such as exhaustive search on the circuit to obtain full coverage of the design space, with almost no time consumed in the optimization process.

From the previous study for the ring oscillator a Latin Hypercube Sampling (LHS) of the circuit with 50 points is selected to create an accurate metamodel [12]. The polynomial metamodel has a root mean square error (RMSE) of 20 MHz with a maximum standard

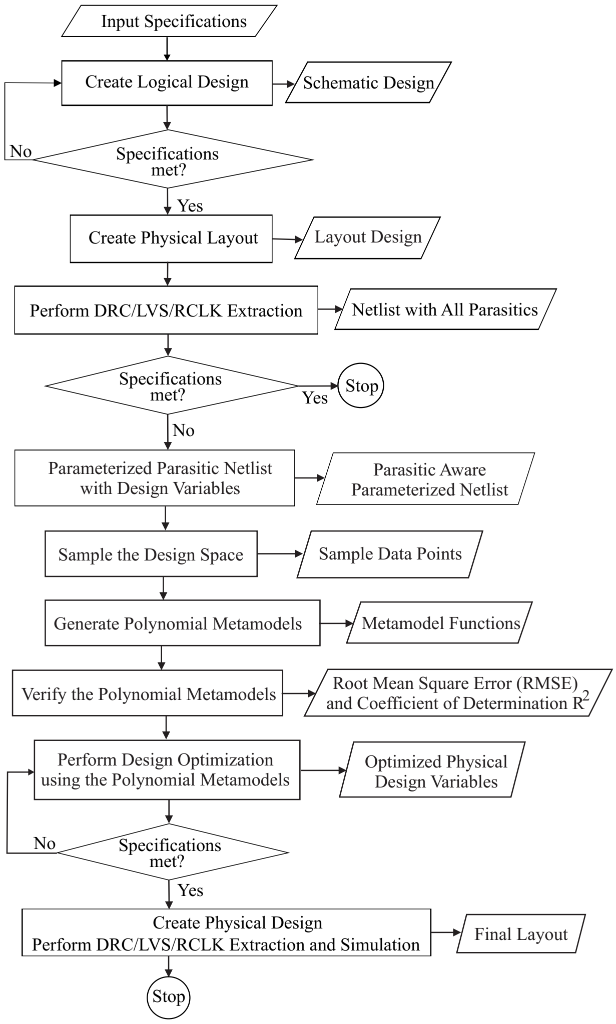

Input Specifications
Create Logical Design
Schematic Design
No
Specifications met?
Yes
Create Physical Layout
Layout Design
Perform DRC/LVS/RCLK Extraction
Netlist with All Parasitics
Specifications met?
Yes
Stop
No
Parameterized Parasitic Netlist with Design Variables
Parasitic Aware Parameterized Netlist
Sample the Design Space
Sample Data Points
Generate Polynomial Metamodels
Metamodel Functions
Verify the Polynomial Metamodels
Root Mean Square Error (RMSE) and Coefficient of Determination R2
Perform Design Optimization using the Polynomial Metamodels
Optimized Physical Design Variables
No
Specifications met?
Yes
Create Physical Design Perform DRC/LVS/RCLK Extraction and Simulation
Final Layout
Stop

Fig. 8 The proposed polynomial metamodel based AMS-SoC design optimization flow.

deviation for error of 10.7 MHz. With the output frequency being in the 10 GHz range, the accuracy of the metamodel is approximately 99.9%. As for the LC-VCO, metamodels were generated with order 1 through 6. The 5th order polynomial exhibited the best results for that circuit with RMSE of 16.8 MHz and highest standard deviation of 9.6 MHz. The accuracy of the metamodel is calculated at 99.6%. The RMSE values are calculated from new sampling points which are different from the points used during regression. This ensures that the RMSE calculation is not artificially optimistic.

The polynomial metamodels are generated for power and frequency for the LC-VCO and ring oscillator circuits. The 1st order to 5th order polynomials are investigated. The 2nd order oscillating-frequency metamodels are the following:

LC-VCO – Order 2:

$$\begin{aligned} f(W_n, W_p) &= 2.3 \times 10^9 + 6.25 \times 10^{11} W_n + 1.45 \times 10^{11} W_p \\ &\quad -4.16 \times 10^{16} W_n^2 - 2.01 \times 10^{17} W_n W_p - 1.02 \times 10^{17} W_p^2. \end{aligned} \tag{1}$$

Ring oscillator – Order 2:

$$\begin{aligned} f(W_n, W_p) &= 6.38 \times 10^9 + 2.2 \times 10^{16} W_n + 6.1 \times 10^{15} W_p \\ &\quad -5.03 \times 10^{22} W_n^2 + 3.28 \times 10^{22} W_n W_p - 1.52 \times 10^{22} W_p^2. \end{aligned} \tag{2}$$

6 Algorithms for Physical-Design Optimization

In this section three algorithms are elaborated and their performance is evaluated for different oscillator circuits. Each optimization algorithm uses two variables: W_n and W_p, for the widths of NMOS and PMOS, respectively. The criteria for comparing the algorithms consist of the following: (1) the time elapsed to do the search, (2) the number of iterations that each algorithm performed while searching, (3) the number of sampling points, (4) whether the number of iterations are different than the number of sampling points (in the case of tabu search), and (5) the accuracy of the result. In the following discussion, all algorithms use S_i for an objective function that holds the best result and S_i^* for an objective function that stores temporary results during the iterative optimal solution search process.

6.1 Algorithm-ESO : Based on Exhaustive Search Optimization

The exhaustive search optimization approach is described in Algorithm 1. This should only be used if simulation time is not an issue. However, in this paper this is considered as a baseline algorithm for the purpose of comparison. It is simple to implement and provides the designer with all available results for all searched points within the constrained space.

Algorithm 1 ESO: Exhaustive Search Optimization

```
Determine the figure of merit (FoM) of interest for the mixed-signal component.
Determine the tuning parameters for the mixed-signal component.
Determine the step size $Step$ of each variable between $W_{nmax}$, $W_{nmin}$ and $W_{pmax}$, $W_{pmin}$ for $N$ simulations.
Initialize the result counter $result\_counter = 0$.
for ($i \leftarrow W_{nmin}$ to $W_{nmax}$ with $Step_{W_n}$) do
    for ($j \leftarrow W_{pmin}$ to $W_{pmax}$ with $Step_{W_p}$) do
        Calculate the FoM of the mixed-signal component using its polynomial metamodel.
        Assign the current objective $S_{ij} \leftarrow$ Calculate from the polynomial metamodel.
        if (The objective is is within the error margin) then
            $result[result\_counter] \leftarrow S_{ij}$.
            Increment the $result\_counter$.
        end if
    end for
end for
return $result$.
```

In the current case, the exhaustive search is conducted for various numbers of iterations taking into consideration the widths of PMOS and NMOS as variables. The design points

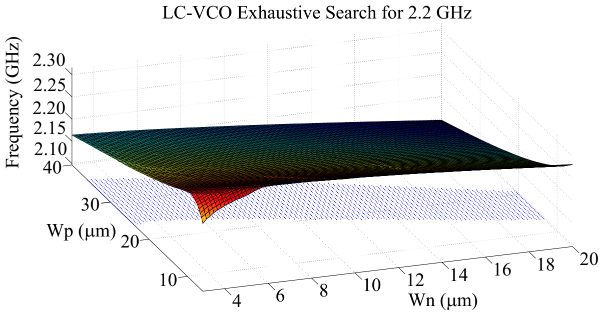


Fig. 9 100×100 search for 2.2 GHz design specification of the LC-VCO. The top surface is the golden surface for the complete design space exploration. The bottom surface (on the $x-y$ plane) is the search result.

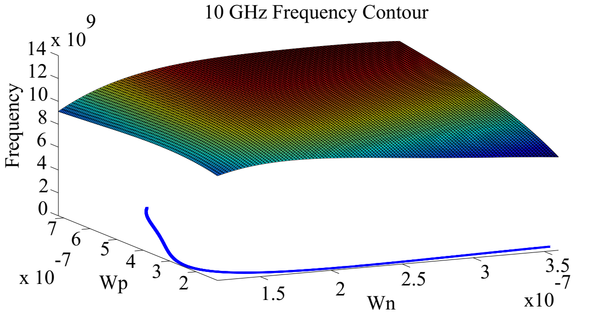


Fig. 10 1000×1000 search for 10 GHz design specification of the ring oscillator. The top surface is the golden surface for the complete design space exploration. The bottom curve is the search result.

that the ESO algorithm determined for the LC-VCO are shown in Fig. 9 and in Fig. 10 for the RO.

Let us assume that there are n number of design/tuning variables for mixed-signal component and each variable is divided into v amount of steps. The ESO algorithm has time complexity of $\Theta(n^v)$.

6.2 Algorithm-SAO: Based on Simulated Annealing Optimization

Simulated annealing optimization is an extension of the Monte Carlo technique that simulates the annealing process in metals [35,28]. Hence it has a random component and consequently two successive runs will produce different results. The steps of simulated annealing based search are presented in Algorithm 2.

The SAO algorithm action as it searches a 10 GHz frequency for the RO in the constraint space is shown in Fig. 11. The starting point is chosen to be in the middle, and the step value is a random value which is normally distributed between 0 to $0.1W_{nmax}$ on the x-axis and 0 to $0.1W_{pmax}$ on the y-axis. The algorithm's temperature setting does not affect its performance for this particular circuit since it is very smooth, as shown in Fig. 11. However, it is usually set as a high number and needs to be able to reach 0 when the inner loop completes its

Algorithm 2 SAO: Simulated Annealing Optimization

```
Determine the figure of merit (FoM) of interest for the mixed-signal component.
Determine the tuning parameters for the mixed-signal component.
Initialize iteration counter $Counter \leftarrow 0$.
Initialize feasible solution $S_i \leftarrow (mid(W_n), mid(W_p))$.
Determine initial FoM for the solution $S_i$ i.e. $\text{FoM}_{S_i}$ using the polynomial metamodels.
Initialize temperature $T$ as $T_i$.
while ($\text{FoM}_{S_i}$) is not within error margin) do
    $Counter \leftarrow$ Maximum number of iterations.
    while ($Counter > 0$) do
        Generate random transition from $S_i$ to $S_i^*$.
        Determine FoM for the solution $S_i$ i.e. $\text{FoM}_{S_i}$ using the polynomial metamodels.
        Determine FoM for the solution $S_i^*$ i.e. $\text{FoM}_{S_i^*}$ using the polynomial metamodels.
        if ($S_i^*$ is acceptable solution) then
            Update the result with $S_i^*$, i.e. $S_i \leftarrow S_i^*$.
            break both while loops.
        else
            Calculate change as: $\Delta FoM \leftarrow \text{FoM}_{S_i}$ - $\text{FoM}_{S_i^*}$.
            if ($\Delta FoM < 0$ and random(0,1) $< e^{\frac{\Delta FoM}{T}}$) then
                Update the solution with new solution: $S \leftarrow S_i^*$.
            end if
        end if
        Decrement $Counter$.
    end while
    Decrease temperature as: $T \leftarrow T \times Cooling\_Rate$.
end while
return the $result$ $S_i$.
```

cycles. The max cycle time is set to 50 iterations for the inner loop with T = 100 degrees and according to that, $Cooling\_Rate = (max\_iteration/T) - 1$, which in this case is 0.5.

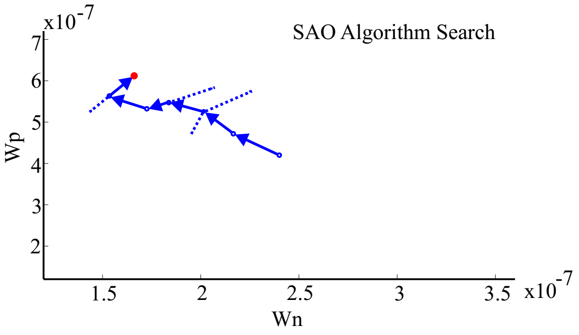


Fig. 11 SAO algorithm search for 10 GHz objective for the RO. The search progressed in the direction of the arrows.

Let us assume that the algorithm goes through n number of iterations. The transition solution space is a constant time complexity step as obtained from DOE analysis. Let us assume that the cooling rate is γ. The time complexity of the SAO algorithm is calculated as $O\left(\frac{n}{\gamma}\right)$.

6.3 Algorithm-DOE-TSO: Based on Design of Experiments (DOE) and Tabu Search Optimization

The DOE-TSO steps are described in Algorithm 3. This is a meta-heuristic algorithm that takes a more aggressive approach than most other search approaches. It skips inferior solutions other than the cases when it needs to exit out of a local optimum [31]. This algorithm uses the entire search constrained space and applies the divide and conquer approach. In the case of tabu search both design variables yield more than one result, as the algorithm chooses more than one point.

Algorithm 3 DOE-TSO: Combined Design of Experiments (DOE) and Tabu Search Optimization.

```
Determine the figure of merit (FoM) of interest for the mixed-signal component.
Determine the tuning parameters for the mixed-signal component.
Initialize iteration counter $Counter \leftarrow 0$.
Perform design of experiments analysis for $W_n$ and $W_p$ using a center design with 9 points.
Generate initial solution $S_i$.
Calculate $\mathrm{FoM}_{S_i} \leftarrow$ from the polynomial metamodel functions.
while ($Counter < Max\_Counter$) do
    Perform Design of Experiment (DOE) analysis to generate a set of 5 points with resulting FOMs
    calculated from the polynomial metamodels.
    Generate the next feasible solution in the selected quadrant as $S_i^*$.
    Increment $Counter$.
    if ($S_i$ is not visited in the previous iterations) then
        if ($\mathrm{FoM}_{S_i^*}$ is better solution than $\mathrm{FoM}_{S_i}$) then
            if (result is found) then
                Update result with $S_i^*$, i.e. $S_i \leftarrow S_i^*$.
                break while loop.
            end if
            Update result with $S_i^*$, i.e. $S_i \leftarrow S_i^*$.
        else
            Discard the solution $S_i^*$ as it is an inferior solution.
        end if
    end if
end while
return the result $S_i$.
```

The algorithm performs a design of experiments (DOE) analysis of the search space. The search space is recursively divided into four adjacent subspaces. Analysis of each of those areas yields the best area. The idea is demonstrated in Fig. 12. The best area is shown in green color and the yellow area is then discarded. The algorithm moves to that area by constraining to its boundaries and then conducts the same analysis of that subspace. In that process the algorithm is able to perform a detailed search in the region that the result is potentially present. Fig. 12 shows the points that the algorithm has sampled to find the value within the search constrained space. The red dot shows the final result of the search for the 9 GHz frequency, which completed in 4 iterations of the algorithm loop and uses 30 sampling points to obtain this result.

Let us assume that the maximum number of iterations in the DOE-TSO algorithm is n. During each iteration the algorithm performs DOE for fixed set of points using the polynomial metamodels, thus taking constant time. Hence, it is concluded that the DOE-TSO has time complexity of $O(n)$.

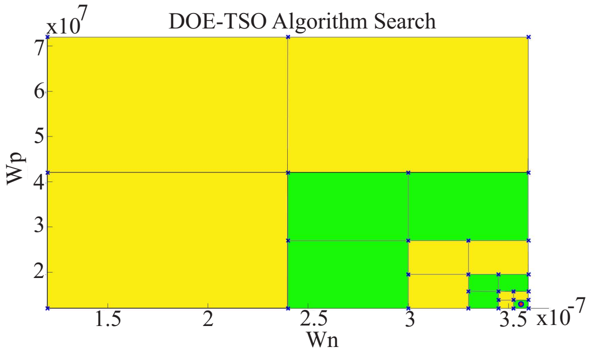


Fig. 12 DOE-TSO algorithm search for 9 GHz of the RO. The search space is recursively divided into rectangles and each time the rectangle with superior result is selected; the other 3 with inferior solutions are discarded.

6.4 Algorithm-ABC: Based on Artificial Bee Colony Optimization

The artificial bee colony (ABC) algorithm is a recently invented population-based metaheuristic approach [3,36,19]. The ABC algorithm mimics the intelligent foraging behavior of honey bee swarms. In the ABC algorithm, the colony of virtual bees is classified into three types: workers, onlookers, and scouts. The key working principle of the ABC algorithm is that a swarm of artificial bees is generated and moved randomly. The artificial bees interact when they obtain some nectar. The solution of a specific problem under consideration is obtained from the intensity of the interactions of these artificial bees. The number of worker bees is the number of solutions of the population. The quality of the solution is related to the nectar amount. A solution of the optimization problem is essentially a position of the food source. The ABC optimization starts with a random initial solution i.e. food source for all worker bees and the algorithm iterates the following steps while requirements are met. Each worker bee goes to a food source and evaluates its nectar amount. Each onlooker bee watches the dance of worker bees and chooses one of their sources depending on the dances and evaluates its nectar amount. Each scout bee determines abandoned food sources and replaces them with new food sources discovered by scouts. The actions of the bees in the context of oscillator circuit optimization is depicted in Fig. 13.

As evident from Fig. 13, the movement of the worker, onlooker, and scout bees are dependent on two factors: the probability of food source ($\mathrm{Prob_{food}}$) and the figure-of-merit (FoM) of the oscillator. The worker bee continues its evaluation as long as the current FoM exceeds the previous FoM, i.e. $\mathrm{FoM_{curr}} > \mathrm{FoM_{prev}}$. Otherwise the transition is made to an onlooker bee. The onlooker bee continues if $\mathrm{Prob_{food}}$ is low. When $\mathrm{Prob_{food}}$ is high and $\mathrm{FoM_{curr}} > \mathrm{FoM_{prev}}$, a transition is made to worker bee. When $\mathrm{Prob_{food}}$ is high and $\mathrm{FoM_{curr}} < \mathrm{FoM_{prev}}$, a transition is made to scout bee. The scout bee continues its evaluation as long as the next FoM does not exceed the current FoM, i.e. $\mathrm{FoM_{curr}} < \mathrm{FoM_{prev}}$. The transition from scout bee to worker bee is made when $\mathrm{FoM_{curr}} > \mathrm{FoM_{prev}}$. A detailed pseudocode of the ABC based solution approach is presented in Algorithm 4 [3,19].

In the ABC steps presented in Algorithm 4, the initial food source is calculated using the following expression [3,19]:

$$d_{j,k} = d_{min,k} + rand(0,1)\left(d_{max,k} - d_{min,k}\right), \tag{3}$$

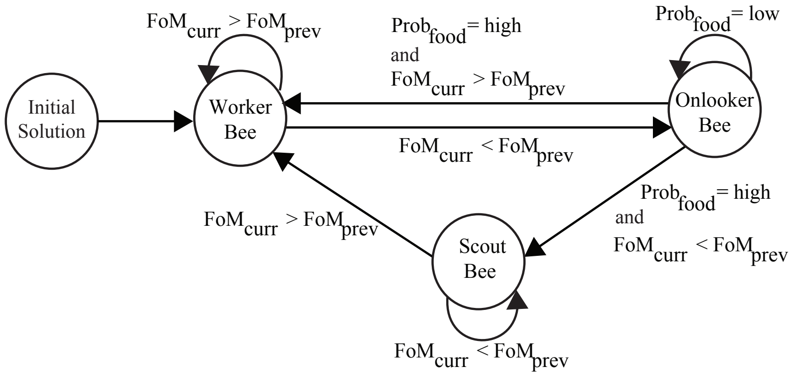

Initial Solution
Worker Bee
Onlooker Bee
Scout Bee
$\text{FoM}_{\text{curr}} > \text{FoM}_{\text{prev}}$
$\text{Prob}_{\text{food}}$= high and $\text{FoM}_{\text{curr}} > \text{FoM}_{\text{prev}}$
$\text{Prob}_{\text{food}}$= low
$\text{FoM}_{\text{curr}} < \text{FoM}_{\text{prev}}$
$\text{FoM}_{\text{curr}} > \text{FoM}_{\text{prev}}$
$\text{Prob}_{\text{food}}$= high and $\text{FoM}_{\text{curr}} < \text{FoM}_{\text{prev}}$
$\text{FoM}_{\text{curr}} < \text{FoM}_{\text{prev}}$

Fig. 13 The transition is states of the artificial bees in the beehive implemented in the artificial bee colony algorithm.

where d is any design, j is any integer from 1 to the number of food sources (i.e. sample points), k is any integer from 1 to the number of design variables, and $rand(0,1)$ generates a uniform random number between 0 and 1. The same expression can be used to generate any random food source. This expression is a constant time consuming operation in the algorithm. An onlooker bee chooses a food source based on the probability associated with that food source, which is calculated using the following expression:

$$\text{Prob}_{\text{food},j} = \left(\frac{\text{fitness}_j}{\sum_{j=1}^{\text{\# of food sources}} \text{fitness}_j} \right). \quad (4)$$

The time complexity of the $\text{Prob}_{\text{food}}$ computation for all food sources is $O(s)$, if s is the number of food sources. Let us assume that the number of design variables, which is same as the number of bees, is n. The algorithm goes through m number of iterations. The time complexity of the ABC algorithm is calculated as $O(nms)$ [3]. The ABC algorithm can be said to have linear complexity with respect to the number of design variables i.e. $O(n)$ [3]. The number of iterations m is user dependent and also can be replaced with an error tolerance criterion. The number of food sources s can also be limited by the user.

7 Experimental Results

The algorithms presented in the previous section are now examined through experimental evaluation of the case study circuits. The two tuning parameters, W_n and W_p, are dictated by nano-CMOS technology adopted for the designs. For example, for the 45 nm ring oscillator, 120 nm width is the minimum value for both variables and 360 nm and 720 nm are the maximum values for W_n and W_p, respectively. For the 180 nm LC-VCO the ranges are 3 μm to 20 μm for NMOS and 6 μm to 40μm for PMOS. The optimization is conducted so as to target a certain output, in the current case frequency (f), with 5% accuracy or better.

The performance of the ESO algorithm with an accuracy of 1% is shown in Table 4. For example, the simulation time for 10,000 points of the RO circuit the algorithm took approximately 32 hours to run the actual simulation *on the parasitic netlist* and found 42 points with the same minimum and maximum values. The optimization data on the polynomial metamodels are also shown. It is evident from the table that the search time has been reduced

Algorithm 4 ABC: Artificial Bee Colony Optimization

```
Set the boundaries for each design parameter, e.g. $W_n, W_p, L_n, L_p$.
Initialize Number-of-Bees (i.e. Number of parameters), Bee-Matrix[workers, onlookers, scouts] (i.e. 0/1
  entries), food sources (i.e. sample point).
while (counter $\leq$ Max-Counter) do
  for each $i$ from 1 to Number-of-Bees do
    if ( Bee-Matrix(1,i) == 1 ) then
      Send worker bee to a random known food source.
      Calculate FoM of the oscillator circuit using polynomial metamodels.
      if (current FoM is better than the previous FoM) then
        Update results (FoM of oscillator circuit) and locations (i.e. design parameters).
      else
        Convert worker bee to onlooker bee.
      end if
    else if ( Bee-Matrix(1,i) == 1 ) then
      Send onlooker bee.
      Calculate probability of the food source being ($\mathrm{Prob}_{\mathrm{food}}$) good.
      if ($\mathrm{Prob}_{\mathrm{food}}$ is high) then
        Send onlooker bee to random location for each design parameter and Calculate the FoM of
          oscillator circuit using polynomial metamodels.
        if (current FoM is better than the previous FoM) then
          Update results (FoM of oscillator circuit) and locations (i.e. design parameters).
          Convert onlooker bee to worker bee.
        else
          Convert onlooker bee to scout bee.
        end if
      end if
    else
      Send scout bee and Pick the best result.
      Send the scout bee to random location for each design parameter.
      if (current FoM is better than the previous FoM) then
        Update results (FoM of oscillator circuit) and locations (i.e. design parameters).
        Convert scout bee to worker bee.
      end if
    end if
    if (current FoM is better than previous FoM) then
      Update results (FoM of oscillator circuit) and locations (i.e. design parameters).
    end if
  end for
  Increment counter.
end while
return Update results (FoM of oscillator circuit) and locations (i.e. design parameters).
```

significantly, as the same 10,000 point optimization took only 0.46 seconds. Of course the strength of this approach lies in the fact that an accurate polynomial metamodel is generated by intelligent sampling of the design space.

The performance of the SAO algorithm for 2.2 GHz frequency of LC-VCO with an accuracy of 1% is presented in Table 5. For the LC-VCO, it is observed that the SAO algorithm reaches the optimal solution in approximately 25 iterations, which is within 0.06% of the needed result, within approximately 22 ms. The performance of the SAO for the 10 GHz frequency RO with an accuracy of 5% is presented in Table 6. It is observed that the algorithm reaches the first optimal solution in 10 iterations which is within 0.48% of the needed result within 0.77 ms.

The performance of the DOE-TSO algorithm is presented in Table 7 for the LC-VCO. DOE-TSO algorithm reaches the optimal solution in approximately 3 iterations which is

Table 4 Results of ESO algorithm for 2.2 GHz LC-VCO and 10 GHz Ring Oscillator (RO).

LC-VCO			Ring Oscillator (RO)		
Iterations	Points Found	Time	Iterations	Points Found	Time
Parasitic Netlist Optimization (Without Metamodel)					
10000	2940	47.2 hours	10000	42	32 hours
2000	587	9.5 hours	2500	13	8 hours
1000	300	4.7 hours	625	2	2 hours
Metamodel Optimization					
1000000	10747	3.65 s	1000000	4566	57.01 sec
10000	1070	313.8 ms	250000	1142	21.73 sec
2000	204	80.03 ms	10000	44	0.46 sec

Table 5 Results of SAO algorithm for 2.2 GHz frequency of LC-VCO.

Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization (Without Metamodel)				
34	2.3 GHz	2.31 GHz	0.44%	7.4 min
60	2.2 GHz	2.22 GHz	0.97%	15 min
24	2.1 GHz	2.08 GHz	0.99%	5.6 min
Metamodeling Optimization				
31	2.3 GHz	2.3 GHz	0.06%	23.8 ms
18	2.2 GHz	2.2 GHz	0.09%	18.1 ms
30	2.1 GHz	2.1 GHz	0.03%	25.0 ms

Table 6 Results of SAO algorithm for 10 GHz frequency of Ring Oscillator.

Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization (Without Metamodel)				
35	9 GHz	8.97 GHz	0.33%	6.84 min
24	10.5 GHz	10.40 GHz	0.97%	4.69 min
16	11 GHz	10.96 GHz	0.36%	3.12 min
Metamodeling Optimization				
32	9 GHz	8.96 GHz	0.48%	1.8 ms
18	9.5 GHz	9.41 GHz	0.94%	1.05 ms
10	10 GHz	10.05 GHz	0.48%	0.77 ms

within 0.06% of the needed result within approximately 78.5 ms for the LC-VCO circuit. The performance of the DOE-TSO algorithm is presented in Table 8 for the RO. It is observed from Table 8 that an average of 17 simulations are expected to run for DOE-TSO algorithm to obtain a result within 5% accuracy with an average of 2.25 loop iterations.

The performance of the ABC algorithm for 2.2 GHz frequency of the LC-VCO with an accuracy of 1% is presented in Table 9. The ABC algorithm reaches the optimal solution in approximately 17 iterations which is within 0.91% of the needed result within approximately 40.5 ms for the LC-VCO design. The performance of the ABC algorithm for the 10 GHz frequency RO with an accuracy of 5% is presented in Table 10. For the ring oscillator circuit, the ABC algorithm reaches the optimal solution in approximately 17 iterations which is within 2.2% of the needed result within approximately 30.4 ms.

Table 7 Results of DOE-TSO algorithm for 2.2 GHz frequency of LC-VCO.

Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization (Without Metamodel)				
5	2.3 GHz	2.3 GHz	0.09%	8.3 min
4	2.2 GHz	2.2 GHz	0.31%	7.2min
5	2.1 GHz	2.1 GHz	0.6%	8.3 min
Metamodeling Optimization				
4	2.3 GHz	2.3 GHz	0.02%	85.2 ms
3	2.2 GHz	2.2 GHz	0.17%	65.4 ms
3	2.1 GHz	2.1 GHz	0.06%	84.9 ms

Table 8 Results of DOE-TSO algorithm for 10 GHz frequency of Ring Oscillator.

Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization (Without Metamodel)				
32	9 GHz	9.38 GHz	4.22%	6.25 min
18	10.5 GHz	10.5 GHz	0.32%	3.52 min
10	11 GHz	11.1 GHz	0.84%	1.95 min
Metamodeling Optimization				
30	9 GHz	9.4 GHz	4.41%	8.6 ms
12	10 GHz	9.93 GHz	0.74%	7.18 ms
24	10.5 GHz	10.5 GHz	0.32%	7.38 ms

Table 9 Results of ABC algorithm for 2.2 GHz frequency of LC-VCO.

Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization (Without Metamodel)				
30	2.3 GHz	2.31 GHz	0.43%	9.2 min
25	2.2 GHz	2.21 GHz	0.45%	7.5 min
16	2.1 GHz	2.10 GHz	0.02%	4.3 min
Metamodeling Optimization				
22	2.3 GHz	2.32 GHz	0.87%	52.3 ms
18	2.2 GHz	2.22 GHz	0.91%	41.8 ms
12	2.1 GHz	2.12 GHz	0.95%	27.5 ms

Table 10 Results of ABC algorithm for 10 GHz frequency of Ring Oscillator.

Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization (Without Metamodel)				
32	9 GHz	9.1 GHz	1.11%	7.8 min
26	10.5 GHz	10.60 GHz	0.95%	6.2 min
18	11 GHz	11.2 GHz	1.82%	3.5 min
Metamodeling Optimization				
28	9 GHz	9.2 GHz	2.22%	41.5 ms
16	10.5 GHz	10.7 GHz	1.9%	32.6 ms
12	11 GHz	11.3 GHz	2.7%	20.3 ms

The four algorithms are now compared in terms of their running time when they performed optimization with and without the metamodel. The lower half of each of the results presented in Table 4, Table 7, Table 8, Table 5, and Table 6 have the running time information. The comparative perspective is presented in Fig. 14 and Fig. 15 in order to provide a visual perspective, for the LC-VCO and ring oscillator, respectively. The calculated speedup is shown in Table 11 and 12. It is interesting to note that SAO algorithm has greater speedup amongst metamodel-based optimizations and DOE-TSO performs better with netlist-based optimizations for both case study circuits.

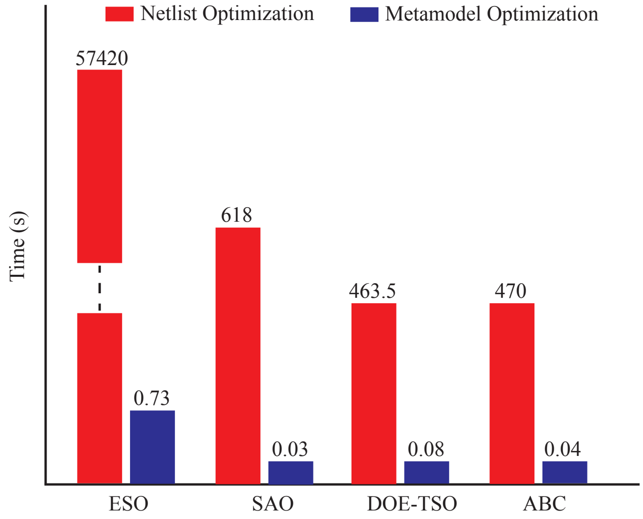


Fig. 14 Performance of the three algorithms for the LC-VCO.

Table 11 Results of the four Optimization Algorithms: Simulation Speedup.

Optimization	Simulation Time (in Sec)				Metamodeling Speedup	
	LC-VCO		Ring Oscillator		LC-VCO	Ring Oscillator
	Netlist	Metamodel	Netlist	Metamodel		
ESO	57,420	0.73	50,400	19.02	78,657×	2,650×
SAO	648	0.03	178	0.02	21,600×	8,900×
DOE-TSO	463.5	0.08	46.8	0.12	5,794×	387.5 ×
ABC	470	0.04	320	0.03	11,750×	10,526×

It is difficult to provide a fair comparison of the proposed research of the current paper with prior research discussed in Section 3. This is due to the fact that the scope and the case study circuits are not exactly the same. From Table 11 it is observed that the speed up is highest (78,657×) for ESO when metamodels are used, however ESO is very time consuming for netlist based optimization. For the 3 metaheuristic algorithms used over netlist and metamodels, it can be stated that the current paper advances the state-of-the art by making the design cycle as fast as $\approx 21,600\times$ for the LC-VCO and $\approx 11,750\times$ for the RO. The

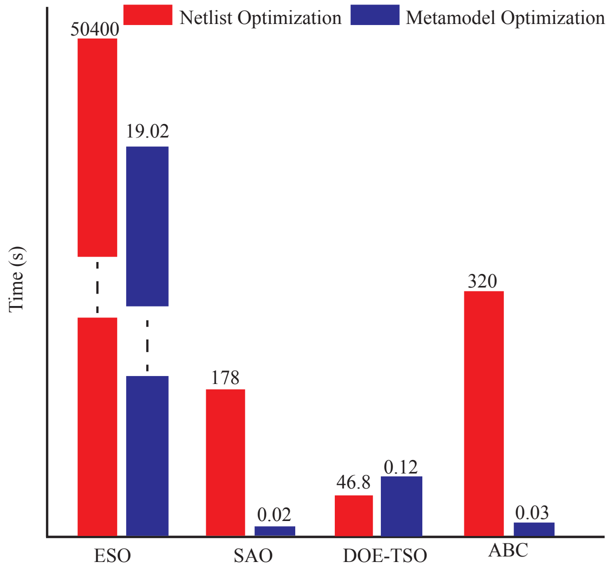


Fig. 15 Performance of the three algorithms for the ring oscillator.

Table 12 Results of the four Optimization Algorithms: Search Speedup.

Optimization Algorithms	Speedup Over Exhaustive Search			
	Netlist (LC)	Metamodel (LC)	Netlist (RO)	Metamodel (RO)
ESO	Baseline	Baseline	Baseline	Baseline
SAO	87×	24.3×	283×	951×
DOE-TSO	124×	9.1×	1077×	158.5×
ABC	122×	18.2×	157.5×	634×

two case-study circuits are carefully selected as they are widely used as oscillators in any AMS-SoC. However, the scalability of the algorithms for complex circuits is crucial. The swarm intelligence based ABC optimization algorithm performs quite well for the LC-VCO as well as the ring oscillator. However, its true potential is more obvious for larger mixed-signal circuits. The polynomial-metamodel based optimization is further investigated using the ABC algorithm for larger circuits such as a phase-locked loop (PLL) with more than 1000 transistors and 21 design variables and it is found that the approach is scalable [13]. The polynomial-metamodel based optimization is further investigated using the SAO algorithm for PLL component circuits and it is found that the approach performs very well [30].

8 Conclusions and Future Research

A novel design flow using metamodels is proposed to minimize the amount of time for AMS-SoC subsystem optimization, with emphasis on oscillators. The fast analysis of four different optimization algorithms demonstrated that ABC, SAO, and DOE-TSO compute the result with fewer iterations than exhaustive search, even though they both stop computing on the first available result within the search criteria, while the exhaustive search computes all available results. It is also observed that DOE-TSO performs better for the simulation approach while SAO performs best for metamodeling. The metamodeling approach is competitively close to the one time optimization in the number of iterations and the accuracy of the result. Optimization techniques that are not computationally tractable, such as exhaustive search can be used on a metamodel. If the designers need to optimize the circuit more than once or need to generate all the optimal solutions for the problem, the metamodeling approach dramatically reduces the design optimization time. Future research will include a more complex analog nano-CMOS circuit metamodel optimization, with a larger number of variables, different types of metamodels, and swarm-intelligence optimization algorithms. The time spent on creating a metamodel is around 98% less than exhaustively searching the design area, which has roughly shortened the design process by 106 hours for the LC-VCO.

Acknowledgments

This research is supported in part by NSF award DUE-0942629. A shorter version of this research is presented at the following double-blind review symposium ISQED [11]. The authors would like to acknowledge Dr. Oleg Garitselov, graduate of the University of North Texas.

References

1. Agarwal, A., Vemuri, R.: Hierarchical Performance Macromodels of Feasible Regions for Synthesis of Analog and RF circuits. In: Proc. Des. Aut. Conf., DAC, pp. 429–435 (2005)
2. Aguirre, M.A., Chavez, J., Torralba, A., Franquelo, L.G.: Analog Design Optimization by Means of a Tabu Search Approach. In: Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 1, pp. 375–378 (1994)
3. Akay, B., Karaboga, D.: A Modified Artificial Bee Colony Algorithm For Real-Parameter Optimization. Information Sciences **192**(0), 120–142 (2012). DOI http://dx.doi.org/10.1016/j.ins.2010.07.015. URL http://www.sciencedirect.com/science/article/pii/S0020025510003336. ¡ce:title¿Swarm Intelligence and Its Applications¡/ce:title¿
4. Basu, S., Kommineni, B., Vemuri, R.: Variation-Aware Macromodeling and Synthesis of Analog Circuits Using Spline Center and Range Method and Dynamically Reduced Design Space. In: Proc. Int. Conf. VLSI Des., VLSID, pp. 433–438 (2009)
5. Boyd, S.D., Kim, S.J.: Geometric Programming for Circuit Optimization. In: Proc. Int. Symp. Physical Design, ISPD, pp. 44–46 (2005)
6. Chang, H., Kundert, K.: Verification of complex analog and rf ic designs. Tech. rep., http://www.designers-guide.com/, Los Altos, CA, USA (2007). URL http://www.designers-guide.com/docs/proc2006.pdf
7. Chou, E.Y., Sheu, B.: System-on-a-Chip Design for Modern Communications. IEEE Circuits and Devices Magazine **17**(6), 12–17 (2001)
8. DasGupta, S., Mandal, P.: An Automated Design Approach for CMOS LDO Regulators. In: Proc. Asia South pac. Des. Aut. Conf., pp. 510–515 (2009)
9. Ding, M., Vemuri, R.: Efficient Analog Performance Macromodeling via Sequential Design Space Decomposition. In: Proc. Int. Conf. VLSI Des., VLSID, pp. 553–556 (2006)
10. Fan, K.T., Li, R., Sudjianto, A.: Design and Modeling for Computer Experiments. Chapman and Hall/CRC, London (2006)
11. Garitselov, O., Mohanty, S.P., Kougianos, E.: Fast Optimization of Nano-CMOS Mixed-Signal Circuits Through Accurate Metamodeling. In: in Proceedings of the 12th IEEE International Symposium on Quality Electronic Design (ISQED), pp. 405–410 (2011)
12. Garitselov, O., Mohanty, S.P., Kougianos, E.: A Comparative Study of Metamodels for Fast and Accurate Simulation of Nano-CMOS Circuits. IEEE Transactions on Semiconductor Manufacturing **25**(1), 26–36 (2012)
13. Garitselov, O., Mohanty, S.P., Kougianos, E., Patra, P.: Bee Colony Inspired Metamodeling Based Fast Optimization of a Nano-CMOS PLL. In: Proceedings of the International Symposium on Electronic System Design (ISED, pp. 6–11 (2011)

14. Ghai, D., Mohanty, S.P., Kougianos, E.: Design of Parasitic and Process Variation Aware RF Circuits: A Nano-CMOS VCO Case Study. IEEE Transactions on Very Large Scale Integration Systems (TVLSI) **17**(9), 1339–1342 (2009)
15. Gielen, G.G.E., Rutenbar, R.A.: Computer-aided Design of Analog and Mixed-Signal Integrated Circuits. Proceedings of the IEEE **88**(12), 1825–1854 (2000)
16. H. Lee and T. Choi and S. Mohammadi1 and L. P. B. Katehi: An Extremely Low Power 2 GHz CMOS LC VCO for Wireless Communication Applications. In: Proc. Eur., p. 3134. Microw. Conf. (2005)
17. Hendrickx, W., Gorissen, D., Dhaene, T.: Grid Enabled Sequential Design and Adaptive Metamodeling. In: Proceedings of the Winter Simulation Conference, pp. 872–881 (2006)
18. Klein, F., Araujo, G., Azevedo, R., Leao, R., dos Santos, L.C.V.: On the Limitations of Power Macromodeling Techniques. In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI, pp. 395–400 (2007)
19. Krishnanand, K.R., Nayak, S.K., Panigrahi, B.K., Rout, P.K.: Comparative Study of Five Bio-Inspired Evolutionary Optimization Techniques. In: Proceedings of the World Congress on Nature Biologically Inspired Computing, pp. 1231–1236 (2009). DOI 10.1109/NABIC.2009.5393750
20. Kundert, K., Chang, H., Jefferies, D., Lamant, G., Malavasi, E., Sendig, F.: Design of Mixed-Signal Systems-on-a-Chip. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **19**(12), 1561–1571 (2000)
21. Lamecki, A., Balewski, L., Mrozowski, M.: Towards Automated Full-Wave Design of Microwave Circuits. In: Proceedings of the 17th International Conference on Microwaves, Radar and Wireless Communications, pp. 1–2 (2008)
22. Liu, B., Deferm, N., Zhao, D., Reynaert, P., Gielen, G.G.E.: An Efficient High-Frequency Linear RF Amplifier Synthesis Method Based on Evolutionary Computation and Machine Learning Techniques. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **31**(7), 981–993 (2012). DOI 10.1109/TCAD.2012.2187207
23. Liu, B., Zhao, D., Reynaert, P., Gielen, G.G.E.: Synthesis of Integrated Passive Components for High-Frequency RF ICs Based on Evolutionary Computation and Machine Learning Techniques. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **30**(10), 1458–1468 (2011). DOI 10.1109/TCAD.2011.2162067
24. Mandal, S.K., Sural, S., Patra, A.: ANN- and PSO-Based Synthesis of On-Chip Spiral Inductors for RF ICs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **27**(1), 188–192 (2008). DOI 10.1109/TCAD.2007.907284
25. Mathaikutty, D., Shukla, S.K.: Metamodeling-Driven IP Reuse for SoC Integration and Microprocessor Design. Artech House (2009)
26. Maulik, P.C., Carley, L.R.: Automating Analog Circuit Design Using Constrained Optimization Techniques. In: Proceedings of the IEEE International Conference on Computer-Aided Design, pp. 390–393 (1991)
27. McConaghy, T., Gielen, G.: Analysis of Simulation-Driven Numerical Performance Modeling Techniques for Application to Analog Circuit Optimization. In: Proceedings of the IEEE International Symposium on Circuits and Systems, (ISCAS), vol. 2, pp. 1298–1301 (2005)
28. Mohanty, S., Velagapudi, R., Kougianos, E.: Physical-Aware Simulated Annealing Optimization of Gate Leakage in Nanoscale Datapath Circuits. In: Proceedings of the Design, Automation and Test in Europe, vol. 1, pp. 1191–1196 (2006). DOI 10.1109/DATE.2006.244046
29. Mohanty, S.P., Kougianos, E.: Polynomial Metamodel-Based Fast Optimization of Nanoscale PLL Components. In: J. Haase (ed.) Models, Methods, and Tools for Complex Chip Design: Selected Contributions from FDL 2012. Springer (2014)
30. Mohanty, S.P., Kougianos, E., Garitselov, O., Molina, J.M.: Polynomial-Metamodel Assisted Fast Power Optimization of Nano-CMOS PLL Components. In: Proceeding of the 2012 Forum on Specification and Design Languages, pp. 233–238 (2012)
31. Mohanty, S.P., Ranganathan, N., Kougianos, E., Patra, P.: Low-Power High-Level Synthesis for Nanoscale CMOS Circuits. Springer Science+Business Media LLC, New York, NY 10013, USA (2008)
32. Park, J., Choi, K., Allstot, D.J.: Parasitic-aware design and optimization of a fully integrated CMOS wideband amplifier. In: Proceedings of the Asia South Pacific Design Automation Conference, pp. 904–907 (2003)
33. Rodoni, L.C., Ellinger, F., Jackel, H.: Ultrafast CMOS Inverter with 4.7 ps Gate Delay Fabricated on 90nm SOI Technology. Electronic Letters **40**(20), 1251–1252 (2004)
34. Samanta, R., Hu, J., Li, P.: Discrete Buffer and Wire Sizing for Link-Based Non-Tree Clock Networks. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **18**(7), 1025–1035 (2010)
35. Sankaran, H., Katkoori, S.: Simultaneous Scheduling, Allocation, Binding, Re-ordering, and Encoding for Crosstalk Pattern Minimization during High Level Synthesis. In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI, pp. 423–428 (2008). DOI 10.1109/ISVLSI.2008.95

36. Saranya, P.K., Sumangala, K.: ABC Optimization: A Co-Operative Learning Approach to Complex Routing Problems. Progress in Nonlinear Dynamics and Chaos **1**, 39–46 (2013)
37. Saraswat, D., Achar, R., Nakhla, M.S.: Fast Passivity Verification and Enforcement via Reciprocal Systems for Interconnects With Large Order Macromodels. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on **15**(1), 48–59 (2007)
38. Shukla, S.K.: Metamodeling: What Is It Good For? IEEE Design & Test of Computers **26**(3), 96 (2009)
39. Thain, W.E., Connelly, J.A.: Simulation and modeling-an improved VCO macromodel. IEEE Circuits and Devices Magazine **8**(4), 8–16 (1992)
40. Wenhui, Z., Xinliang, L.: Gaussian Process Meta-Modeling and Comparison of GP Training Methods. In: Proceedings of the International Conference on Logistics Systems and Intelligent Management, vol. 2, pp. 1193–1199 (2010). DOI 10.1109/ICLSIM.2010.5461149
41. Wong, H.S.P., Frank, D.J., Solomon, P.M., Wann, C.H.J., Welser, J.J.: Nanoscale CMOS. Proceedings of the IEEE **87**(4), 537–570 (1999)
42. Wong, J.L., Davoodi, A., Khanderwal, A., Srivastava, A., M. Potkonjak, M.: A Statistical Methodology for Wire-Length Prediction. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **25**(7), 1327–1336 (2006)
43. Xie, L., Davoodi, A., Zhang, J., Wu, T.H.: Adjustment-Based Modeling for Timing Analysis Under Variability. IEEE Trans. on CAD of Integrated Circuits and Systems **28**(7), 1085–1095 (2009)
44. Xu, Y., Hsiung, K.L., Li, X., Nausieda, I., Boyd, S., Pileggi, L.: OPERA: Optimization with Ellipsoidal Uncertainty for Robust Analog IC Design. In: Proc. Des. Aut. Conf., DAC, pp. 632–637 (2005)
45. Xu, Y., Luo, Z., Chen, Z., Li, X.: Average leakage current macromodeling for dual-threshold voltage circuits. In: Proceedings of the 12th Asian Test Symposium, pp. 196–201 (2003)
46. Xu, Y., Pileggi, L.T., Bo: ORACLE: Optimization with Resource of Analog Circuits including Layout Extraction. In: Proc. Des. Aut. Conf., DAC, pp. 151–154 (2004)
47. Yelten, M.B., Zhu, T., Koziel, S., Franzon, P.D., Steer, M.B.: Demystifying Surrogate Modeling for Circuits and Systems. IEEE Circuits and Systems Magazine **12**(1), 45–63 (2012). DOI 10.1109/MCAS.2011.2181095