NOVEL DESIGN METHODS FOR ANALOG DESIGN AUTOMATION TOOLS

by

Gönenç Berkol

B.S., Electrical and Electronics Engineering, Ege University, 2013

Submitted to the Institute for Graduate Studies in

Science and Engineering in partial fulfillment of

the requirements for the degree of

Master of Science

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2015

NOVEL DESIGN METHODS FOR ANALOG DESIGN AUTOMATION TOOLS

APPROVED BY:

Prof. Günhan Dündar   . . . .
(Thesis Supervisor)

Assist. Prof. İsmail Faik Başkaya   . . . .
(Thesis Co-supervisor)

Assoc. Prof. Ali Emre Pusane   . . . .

Assist. Prof. Berke Yelten   . . . .

Assoc. Prof. Arda Deniz Yalçınkaya   . . . .

DATE OF APPROVAL: 10.08.2015

ACKNOWLEDGEMENTS

I would like to express my profound sense of gratitude to Professor Günhan Dündar, my supervisor, for his continuous support, motivation, and admirable knowledge. His guidance encouraged me in all the time of research during my master thesis. His inspiration is the key driving force for me to continue my academic career. Also, I would like to express my deep appreciation and sincere gratitude to Assoc. Prof. Ali Emre Pusane and Assist. Prof. İsmail Faik Başkaya for their invaluable support and guidance. I would like to thank to Assoc. Prof. Arda Deniz Yalçınkaya and Assist. Prof. Berke Yelten for their participation in my thesis jury.

I would like to give special thanks for Engin Afacan, who has made contributions in every part of my thesis. He has been not only a great mentor, but also a big brother to me. I would not be able to succeed without his support. I would like to thank all the wonderful people of BETA for their true friendship and support.

Finally, I owe my gratitude to my family for their continuous love and support.

ABSTRACT

NOVEL DESIGN METHODS FOR ANALOG DESIGN AUTOMATION TOOLS

Improvements in fabrication processes have enabled designers to benefit from sub-micron devices, which have led to the integration of multibillion transistors. However, circuit complexity increases together with the silicon complexity, since the effect of process variations in sub micron technologies are drastically increased. Thus, design automation tools have emerged to reduce design time without sacrificing performances. To achieve that goal, simulation based optimization algorithms for analog circuit circuits are developed in this thesis. In order to synthesize a robust circuit, yield is defined as a new design objective, which is tried to be maximized during the optimization process. To enhance the efficiency of yield estimation, Quasi-Monte Carlo (QMC) method, has been utilized in optimization. In addition, since QMC is deterministic and has no natural variance, there is no convenient way to obtain error bounds for the estimation. To determine the confidence interval of the estimated yield, scrambled-QMC method and conventional QMC method are combined. Therefore, a hybrid method is proposed, where a single QMC is performed to determine infeasible solutions in terms of yield, which is followed by a few scrambled QMC analyses providing variance and confidence interval of the estimated yield. In addition to the variation-aware tools, a layout-aware tool is proposed, in which a simulation-based circuit sizing tool with a template based layout generation tool are combined. The layout-induced parasitics are automatically extracted and are taken into account during the optimization process. To reduce the run time cost due to parasitic extraction, a two step methodology is developed. Finally, the circuits obtained at the end of optimization are implemented as an integrated circuit to show the effectiveness of the algorithms.

ÖZET

ANALOG TASARIM OTOMASTONU İÇİN ÖZGÜN TASARIM YÖNTEMLERİ

Fabrikasyon süreçlerindeki gelişmeler mikron altı cihazların kullanımına olanak sağlamış, böylece milyarlarca transistörün birleştirilmesi gerçekleşmiştir. Fakat, hem devre hem de silikon karmaşıklığı giderek artmaktadır çünkü süreç değişimlerinin etkisi mikron altı teknolojilerde çok fazladır. Bu nedenle, tasarım otomasyonu araçları devre performanslarını karşılayacak ve tasarım süresini kısaltacak şekilde gelişmektedir. Bu amacı gerçekleştirmek üzere bu tezde, analog devreler için benzetim temelli eniyileme algoritmaları geliştirilmiş ve uygulanmıştır. Gürbüz devreler tasarlamak için, verim yeni bir tasarım amacı olarak tanımlanmış ve eniyileme esnasında arttırılması amaçlanmıştır. Daha etkili verim hesabı yapabilmek için Quasi-Monte Carlo (QMC) yöntemiden eniyileme sırasında faydalanılmıştır. Fakat bu yöntemde kararlı sayı dizileri kullanılmasından ötürü verimin varyansı elde edilememektedir, dolasıyla tahmin hatasının sınırları hesaplanamamaktadır. Tahmin edilen verimin güvenilir aralığını elde etmek için karıştırılmış QMC ve standart QMC yöntemleri birleştirilmiştir. Dolayısıyla, melez bir yöntem önerilmiştir. Bu yöntemde önce standart QMC ile verim tahmini yapılmış, daha sonra karmaşık QMC yönteminden faydalanılarak varyans ve güvenilir aralık hesaplanmıştır. Değişim farkındalı araçlara bir ek olarak, benzetim tabanlı devre ölçeklendirme aracı ile şablon tabanlı serim üretim aracı birleştirilerek, serim farkındalı tasarım aracı geliştirilmiş ve önerilmiştir. Serim nedenli devre parazitikleri otomatik olarak çıkartılmış ve eniyileme sırasında kullanılmıştır. Parazitik çıkartmak için gereken süreyi kısaltmak için iki aşamalı bir yöntem geliştirilmiştir. Son olarak eniyileme sonunda elde edilen devrelerin VLSI tasarımları yapılarak, geliştirilen algoritmaların etkililiği gösterilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

CAD	Computer Aided Design
DFM	Design for Manufacturability
DRC	Design Rule Check
EC	Evolutionary Computing
EP	Evolutionary Programming
GA	Genetic Algorithms
GDS	Graphic Data System
HDL	Hardware Description Language
IC	Integrated Circuit
IS	Importance Sampling
ISE	Infeasible Solution Elimination
LHS	Latin Hypercube Sampling
LVS	Layout Versus Schematic
MATLAB	Matrix Laboratory
MC	Monte Carlo
MO	Multi Objective
Op-Amp	Operational Amplifier
OTA	Operational Transconductance
PEX	Parasitic Extraction
PF	Pareto Front
QMC	Quasi Monte Carlo
SA	Simulated Annealing
SO	Single Objective
SoC	System on Chip
SPEA	Strength Pareto Evolutionary Algorithms
SPICE	Simulation Program Integrated Circuit Emphasis
UMC	United Microelectronics Corporation

1. INTRODUCTION

1.1. Motivation

The time period starting from the invention of transistor in 1947 to production of the first integrated circuit (IC) in 1958, can be accepted as the beginning of a new age, which is called the information age. Intelligent electronic products have become indispensable in the daily life for the all kind of people in the society. The evolution of the microelectronic industry is helping societal needs in a number of ways including communication, health, entertainment, energy, security, and education. In this age, especially the interdisciplinary research on bioelectronics and nano-electronics will have a huge impact on people.

The requirement of the high performance and cost effective electronic products led to a trend towards system on chip (SoC). Thanks to the improvements in the fabrication processes that enabled to decrease feature sizes, multi-billion transistors can be combined into a single chip [2]. This achievement has been foreseen by Gordon Moore in 1965 [3], in which he stated that the number of components per integrated circuit would increase exponentially over time. Ever since, increasing the number of transistors per IC has become the driving force for electronic engineers. However, developments in the scaling process and combining many circuits into a single IC comes at a price. In [4], problems are categorized in two groups. The first one is silicon complexity, which implies the effects of process variations on the circuit performance. In deep sub-micron technologies, process tolerances have worsened along with transistor dimensions [5]. Therefore, even if a circuit was designed to achieve a certain design performance, a discrepancy occurs between the expected and the measured performances in a population of fabricated ICs [6], as depicted in the Figure 1.1. The second problem is called system complexity, meaning that exponentially increased number of transistor counts lead to increased functionality. Power and area management of the ICs together with various trade-offs between circuit performances in addition to short time to market demand are the major concerns.
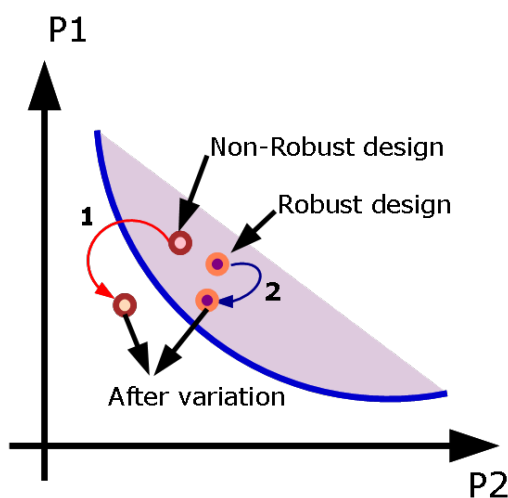
Figure 1.1. Changing of circuit performances after variation.

An efficient way for dealing with design challenges and increasing productivity of designers is to use computer aided design (CAD) tools [2]. CAD tools can provide assistance during analysis and verification of the system. Typically, from transistor level to system level, the designer benefits from CAD tools to determine the performance of the design. In the literature, efficient CAD tools for digital circuits are available. Digital systems are more suitable for design automation contrary to analog systems since digital systems can be defined with Boolean representation. Today's advanced tools are capable of synthesizing a transistor level design that was described in a hardware description language (HDL) [2]. In the perspective of analog systems, developing a CAD tool is more challenging since analog circuits cannot be represented as digital circuits and have more complicated trade-offs between circuit performances and physical parameters. In addition, taking into account second order effects as a result of device scaling, most of the time of analog designers is spent by fine tuning the system performances by utilizing a simulator through trial and error. If, somehow, computer intelligence techniques combining circuit simulators into optimization algorithms are utilized, the overall time spent for the design would be substantially reduced. Also, such a synthesis tool would enable fine tuning of silicon and system

complexities simultaneously. Therefore, the main focus of this thesis is to develop an analog circuit synthesis tool that addresses these problems.

1.2. Analog IC Design Flow

In this section, analog IC design methodologies will be briefly introduced. As explained in the previous section, increased complexity of analog circuits results in growing design productivity gap for SoCs considering the shortened time to market constraint [4]. In order to enhance the design process, some design methodologies are proposed for the designers. In [2], design methodologies are combined into two groups. The first one is the top-down design methodology whose flow diagram is shown in Figure 1.2.
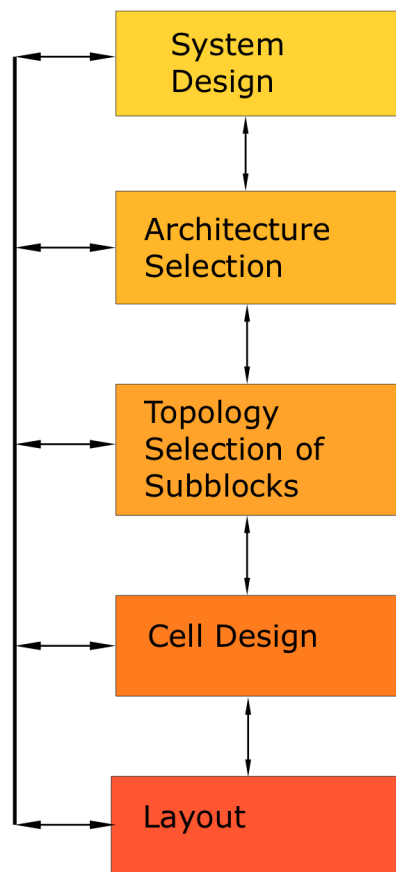
Figure 1.2. Analog IC design flow diagram.

The flow starts with system design, in which overall system requirements are specified. General blocks having dedicated tasks are designed and partitioned into sub-blocks. Typically, mathematical tools such as MATLAB/SIMULINK are preferred for system level design. The next stage is called architecture selection where digital and analog blocks are separated and requirements of functional blocks are defined. The following stage is called topology selection in which topologies for functional blocks and sub-blocks are determined. For example, if an analog to digital converter is required, the designer will determine to use either sigma-delta or flash converter, based on the power or speed constraints. In a similar manner, if an Op-Amp is required, the designer will determine to use either a two-stage or a folded cascode topology. In the cell design, specific blocks are designed at transistor level and sizing is performed to achieve pre-defined performances in a certain technology. Finally, layouts of cells and general blocks are drawn and post-layout simulations are performed to validate circuit and system specifications. This methodology is advantageous because systematic design is suitable for capturing and fixing problems, since it allows interaction of blocks during the design process [2].

The other design methodology, which is called the bottom-up decomposition, in which the designer starts by utilizing previously designed cells [7]. However, using the library of analog cells may be inefficient considering the technology dependency and variety of analog circuits. However, if some form of soft intellectual property (IP) is used, design knowledge and optimization techniques could be embedded such that technology dependency is removed and a wide range of performance choices is provided for designers. Previously generated Pareto fronts of cells, as shown in Figure 1.3, could be used and combined until the desired system performances are achieved [2].

1.3. Objectives of the Thesis

The main idea behind the thesis is to develop an analog circuit synthesis tool for cell design. For a given topology, sizing and biasing of the circuit is carried out by utilizing various optimization algorithms to reach performance specifications, as shown in Figure 1.4. In addition, developing computer intelligence techniques and embedding
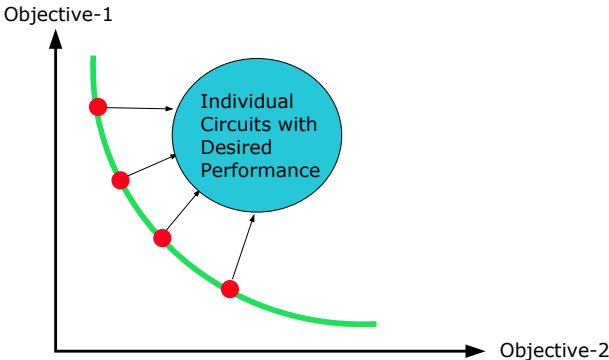
Figure 1.3. Conceptual Pareto front of cell.

them into an analog circuit synthesis tool to improve manufacturability of analog cells are the important driving forces. Yield estimation and optimization along with layout induced parasitic effects, are taken into account to achieve the requirements of the design for manufacturability (DFM).
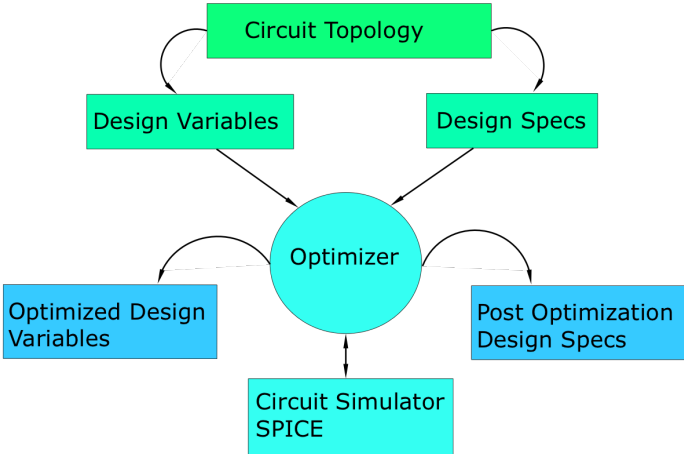
Figure 1.4. Analog circuit synthesis flow for cell design.

1.4. Key features, Contributions and Outline

This thesis presents the following key features and contributions:

- Simulation-based circuit sizing tools given in [8, 9] are revised and implemented in MATLAB,

- Yield estimation techniques are developed and embedded into the tools given above. During the synthesis process, interactions between the yield and the circuit performances are analyzed and the yield is integrated as a new objective in design flow.

- Layout-aware Analog Synthesis Tool is proposed, in which a simulation-based circuit sizing tool [8] and a template based layout generation tool [10] are integrated. Layout-induced parasitic effects are considered during the optimization process in order to satisfy design objectives after the circuits are physically implemented,

- VLSI implementation is performed for the circuits, which are obtained at the end of proposed tool.

The organization of thesis as follows: Chapter 2 presents the background of analog design automation methodologies, evolutionary computation and optimization algorithms. Chapter 3 presents the implementation of the single objective optimization tool and design examples. Chapter 4 explains the implementation of the multi objective tool. Chapter 5 is dedicated to yield-aware design methodologies. Chapter 6 gives the details of the layout-aware tool. Chapter 7 presents test chip Design, layouts, and post-layout simulations. Finally, Chapter 8 concludes the thesis.

2. BACKGROUND

2.1. Overview of Analog Design Automation Tools

Topology selection and circuit sizing are considered as a whole in earlier design automation tools. In [11,12], the circuit schematic is determined before applying sizing algorithms. In this thesis, topology selection is considered to be out of scope, thereby circuit sizing approaches are utilized in proposed tools.

After a certain topology selection process, satisfying pre-defined performance goals is a challenge since careful circuit sizing is required. The design equations that relate performance goals and physical parameters to device sizes require solving mostly high order and complicated mathematical equations. Corresponding to the evaluation methods for design equations and approaches for circuit sizing, knowledge based and optimization based algorithms have been proposed in the literature.

Knowledge based algorithms have been developed at first, in which the designer's experience and design strategies for a given circuit topology is utilized during the circuit synthesis [11,13]. Design plans consisting of design equations and design strategies reduce the computation time required for obtaining solutions. However, preparing design plans for each topology requires excessive human effort [2]. These tools can not have place in the market considering the disadvantages of human interaction in the optimization process and creation time of design plans for various circuit topologies.

Thanks to the advances in computer technology that increases computing capacity, optimization based algorithms are proposed for circuit sizing. These algorithms are categorized into two groups in [2]. The first one is called equation based algorithms. Despite the requirement of design equations, improvements are achieved in the solution strategies by performing optimization algorithms, such as simulating annealing [14]. Thereby, tools given in [15] [16] provide fast convergence rate and flexibility for carrying out various search algorithms. However, design equations still had

to be derived by hand, thereby requiring human interaction. Another disadvantage is the loss of accuracy since it is difficult to derive all design equations without making simplifications.

The second type of optimization algorithms is called simulation based algorithms since commercially available circuit simulators are exploited to measure circuit performances [17, 18]. As a result, human interaction during circuit synthesis for creating design equations is eliminated. In addition, design automation tools overcome the loss of accuracy and become comparable with manual designs. Furthermore, topology and technology dependency is no longer valid since they can be easily manipulated at the input file of SPICE. On the other hand, total synthesis time is drastically increased since optimization algorithms require excessive simulations to find the optimum solution. However, this excessive CPU time can be reduced by developing intelligent algorithms and avoiding infeasible solutions [19].

2.2. Overview of Evolutionary Computation Algorithms

Evolutionary computing (EC) has found a wide range of applications for scientists and engineers due to its general and systematic nature [1] [20]. EC is inspired by the biological evolution as stated by the Charles Darwin in 1859 [21]. In [22] and [1] combination of genetic algorithms (GA), evolutionary strategies (ES) and evolution programming (EP) are defined as EC, where the general structure is given in Figure 2.1. The algorithm starts with arbitrarily generation of the initial population. Population is used to represent candidate or probable design solutions for the optimization problem. It may consist of one dimension but generally consists of more than one dimension. For example, in the analog design automation tool given in [5], a candidate in the population consists of 12 parameters which are transistor dimensions, bias current, and compensation capacitance. After the initialization of the population, parents are selected with respect to performance or randomly at first iteration. Selection is required for choosing the candidates that are going to participate in recombination to produce offspring and thus the next generation. The reason of this operation can be explained in a same way as in the biological evolution. Some of the candidates

in the population are better than the other ones. Better implies to cost or fitness function which indicates that a particular candidate has a high probability to survive and achieve design specifications; thereby, it has to convey its properties to the next generation. However, repeatedly choosing better candidates carries a risk of getting stuck at a local minimum. Thus, elitist selection should be avoided in order to increase diversity of the population. Also, mutation operator is usually performed to further increase diversity.
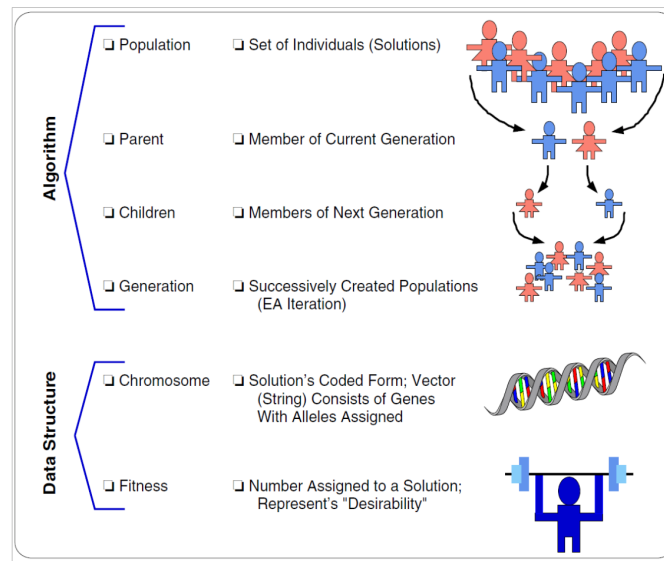
Figure 2.1. General structure of evolutionary computing(from [1]).

In the perspective of analog design automation tools, EC is a popular method since it offers a set of candidate solutions at the end of optimization. Therefore, Pareto optimal solutions are obtained in a single run of the algorithm. Also, these methods are independent of the shape or discontinuity of the Pareto Front (PF) [23].

In general, EC can be utilized and embedded into single objective (SO) and multi objective (MO) optimization algorithms. SO optimization aims to find a global minimum or maximum point for the objective function, which is generally called cost. If the number of design objectives is more than one, the cost consists of the combination of the design objectives into a single function [1, 23]. On the other hand, in MO optimization, the design objectives are independent and equally important, which

allows interaction among them and provides a Pareto front (PF) including all feasible solutions. Therefore, the designer has the flexibility to choose candidates in the PF, considering the trade-offs between them. The output of SO optimization offers solutions that only satisfy pre-determined performance constraints, thereby solutions are not correlated with each other. As a result, the answer to the question of which optimization algorithm is more effective, depends on the application and designer's choice. In addition, it is stated in [23] that, if the number of objectives is two, MO algorithms outperform the SO algorithms. However, when the number of design objectives is increased, SO algorithm is shown to be more effective. As a result, both optimization algorithms are utilized and implemented in this thesis.

3. SINGLE OBJECTIVE OPTIMIZATION

The number of design objectives of an analog design automation problem is generally more than one. As discussed in Chapter 2, when design objectives are lumped into the single function, SO optimization algorithms combined with the EC yields promising results [1, 7, 19]. Therefore, the modified version of the tool given in [8] is implemented in MATLAB and utilized in this thesis.

3.1. Algorithm Implementation

The tool starts with the preparation of the SPICE netlist file, where flow diagram is given in Figure 3.1. Since a certain topology is chosen before, circuit schematic along with the performances to be measured should be translated into a SPICE input file. This approach is useful in particular when the designer decides to change the technology or the topology since the rest of the algorithm is independent of them. Also the designer should introduce good and bad limits for the design objectives to the tool. The next step is the population initialization in which the designer should determine the boundaries of the search space. Population size is set by determining the number of parents and offspring by the designer. Dimensions of the population can be changed corresponding to the topology. In general, a population consists of the transistor widths and lengths, the bias current or resistor, and the compensation capacitance. Based on the upper and the lower bounds that are set by the designer, initial population is created by using pseudo random sampling in [8]. However, the initialization of the population in this thesis is performed by using a Sobol sequence set of design parameters where the advantages of such a sequence over the pseudo-random populations are discussed in [19].

The next step is performance evaluation in which a commercial SPICE program, HSPICE, is used. Output files of HSPICE are stored in MATLAB during the optimization process. Thereby, simulation based approach is exploited in the algorithm and SPICE level accuracy is obtained with the expense of increased CPU time. The
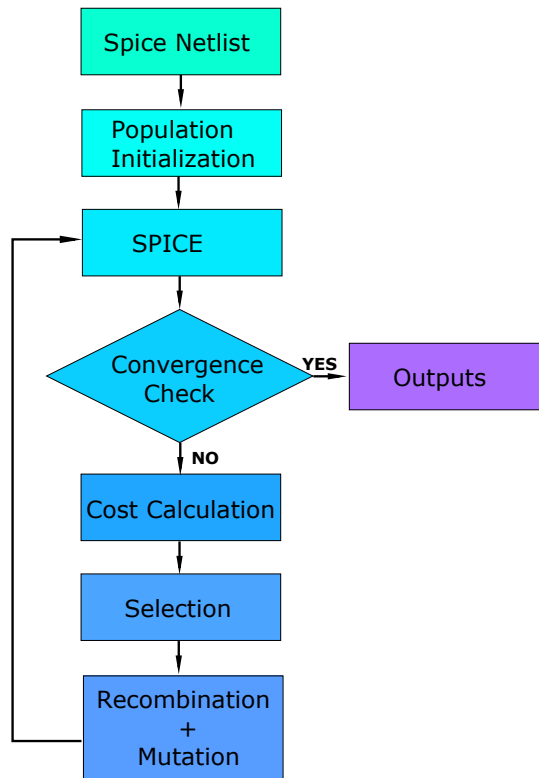
Figure 3.1. Flow diagram of the single objective optimization.

following step is the convergence check, where the tool stops working if the user defined iteration count is reached. In that case, the current population will be the output of the tool. Otherwise, cost calculation is performed to the solutions. The cost is calculated according to

$$Cost = C_{performance} + C_{penalty},$$

$$C_{performance} = \sum_i^n w_i.P_i^2, \tag{3.1}$$

$$P_i = \frac{U_i - f_i}{U_i - L_i}, P_{i,min} = 0,$$

where n, f_i, U_i, and L_i are the number of performance specifications, i^{th} instant, upper,

and lower values of performance metrics, respectively. $C_{penalty}$ is calculated according to the operating point of all transistors, where triode and cut-off regions are penalized. In addition to the search parameters, cost function weights (w_i) could be added for increasing the convergence rate as described in [8]. However, cost function weights are not utilized and set to 1 in this thesis since contributions of automatically evolving cost function weights are not observed.

After a cost is assigned to each individual in the population, a selection algorithm is performed where metropolis criteria and simulating annealing (SA) algorithms are combined. In this approach, individuals having lower cost values than the average cost value of the current population are assigned a higher probability to be chosen as the new parents of the next generation but individuals having a high cost still have a chance to be chosen as the new parents. In addition, at the beginning of the population, it is highly probable to choose an inferior individual and this contributes to population diversity. As the number of iterations is increased, the population focuses on a certain region of the search space, fine tuning the variables and this near end stage is rather a local search around the global optimum.

Production of the next generation is performed by recombination. Selected individuals as a parent are randomly chosen and offsprings are formed by crossing over the dimensions of the individuals. The selected two parents produce two offspring which inherit the critical characteristics from their parents. Therefore, the chance of reaching the required circuit specifications is increased by broadening the search space. Also, to further increase the diversity, designer can set a mutation probability, which can be used to select individuals to be mutated. In that case, the dimensions of the selected individuals are all randomly changed within the boundaries of upper and lower limits that designer set in the initialization. The disadvantage is that, there is a probability that an individual having a low cost is mutated. However, observations show that if the overall mutation probably is kept between $0.2 - 0.4$, increased diversity improves the convergence rate. As a result, production of the next population is performed, then their performances are evaluated again, as shown in the Figure 3.1. This iteration is repeated until the maximum iteration number is reached.

3.2. Synthesis Examples

In order to test the developed tool, a two stage OTA , a folded cascode amplifier, and a comparator were chosen as design examples, which are shown in Figure 3.2, Figure 3.3, and Figure 3.4, respectively.
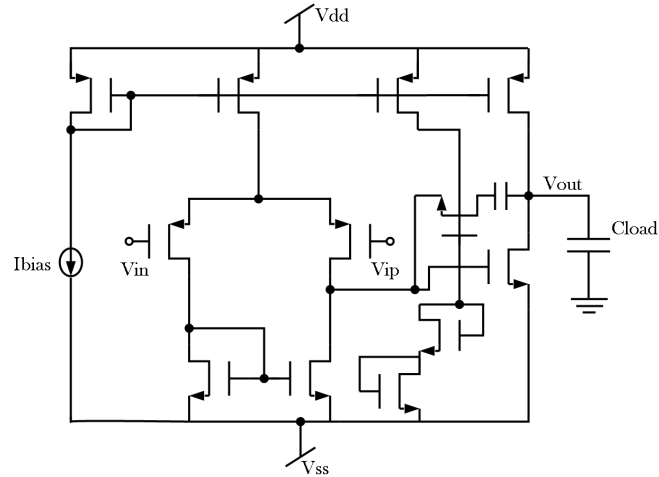
Figure 3.2. Two stage operational transconductance amplifier.
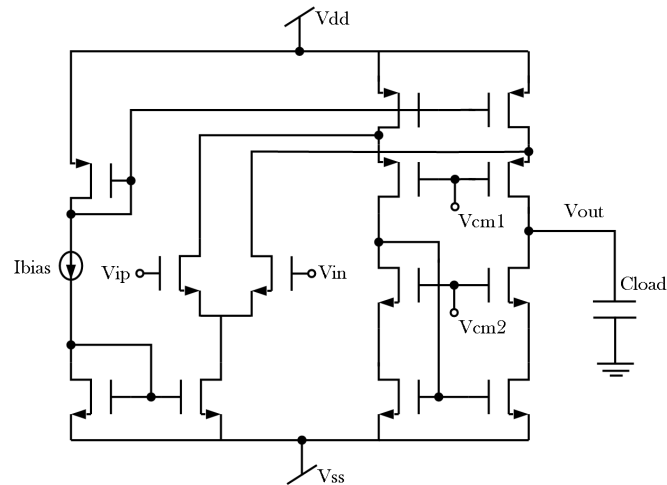
Figure 3.3. Folded cascode amplifier.

All circuit elements such as transistor dimensions, bias voltages, and currents, etc. were given to the optimizer as design parameters, which is 22 for the two stage OTA and 16 for the folded cascode amplifier. Simulations were performed using 130nm

Table 3.1. Synthesis results for 3 independent runs.

Two Stage OTA Results					
	BW(kHz) >15kHz	Gain(dB) >55dB	PM(^{o}) >55°	Power(mW) <1.5mW	Area($e^{-9}m^2$) <1e-9m^2
1	21	68.6	56.5	0.9	0.374
2	22.3	62	86	1.4	0.97
3	17.7	70	75	0.59	0.518
Folded Cascode Amplifier Results					
	BW(kHz) >20kHz	Gain(dB) >55dB	PM(^{o}) >60°	Power(mW) <1.5mW	Area($e^{-9}m^2$) <1e-9m^2
1	137.5	61	63	0.93	0.5
2	92.9	61	60	1.5	0.64
3	78.7	60	67	1.2	0.36

UMC technology and $1.2V$ supply voltage. The optimization process takes $15-20$ minutes with an Intel i7 chipset with 2.80GHz processor. Population size of parents and offspring, and maximum number of iterations were chosen as 200, 200, and 20, respectively. Bandwidth, gain, phase margin, power, and area are chosen as design objectives for both circuit topologies. Required design objectives together with synthesis results of each circuit topology are presented in Table 3.1 for 3 independent runs. It can be concluded that all design objectives are successfully achieved in each run.

Comparators are another example of fundamental analog circuits. Optimization of comparators is significant especially in hierarchical design methodologies since system level designs such as flash ADCs, can be created by combining several comparators. Therefore, in addition to the amplifier topologies, comparator topology given in Figure 3.4 is used in the proposed design automation tool [24].

The major difficulty of optimizing comparators is to determine the offset during circuit synthesis. In general, reference and a ramp voltage input is used to test comparators. Offset sensitivity is determined by observing the input difference that
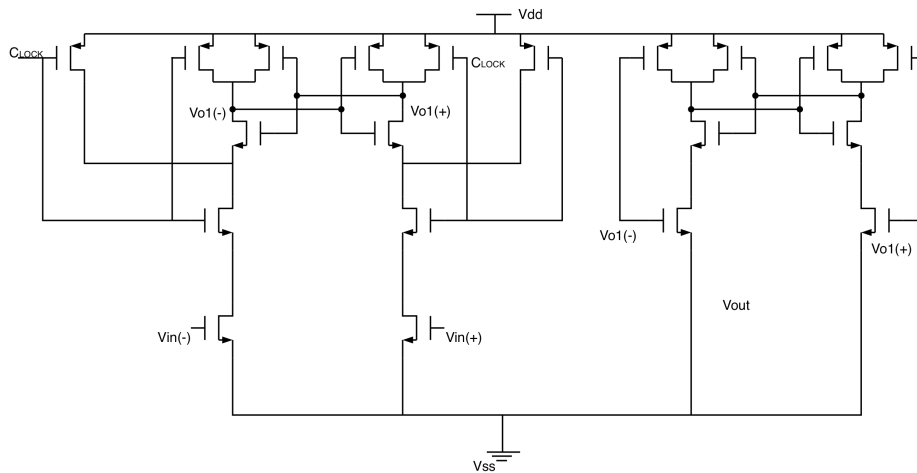
Figure 3.4. Circuit schematic of comparator.

changes the output of comparator. However, this method requires very large samples to measure offset, thereby consumes excessive CPU time. To overcome this bottleneck, a test setup at the input file of the SPICE is developed, in which user defined clock frequency and offset are exploited in simulations. Thus, offset and clock frequency are considered as design constraints in this approach. The positive output of the comparator is sampled at the falling edges of the clock for several cycles. Design constraints are considered to be satisfied if the circuit works properly for a given offset and sampling frequency. Otherwise, it is considered as infeasible and assigned a high cost. In addition, power and area are introduced as design objectives.

Table 3.2. Synthesis results of comparator for 3 independent runs.

	Power(μW) $<300\mu W$	Area($e^{-12}m^2$) $<120e^{-12}m^2$
1	242	100
2	236	75
3	260	90

The results of the synthesis examples for 3 independent runs are given in Table 3.2. Design objectives for power and area are set to $300\mu W$ and $100e^{-12}m^2$, respectively. Sampling frequency is determined as $250MHz$ and a $5mV$ offset is introduced

at the inputs of comparator by applying opposite phase square waves changing from $0.7V$ to $0.695V$. Transient simulations of the outputs of comparator, the clock and the overlapping inputs of first solution are given in Figure 3.5. It can be observed that the resulting comparator satisfies both design objectives and design constraints.
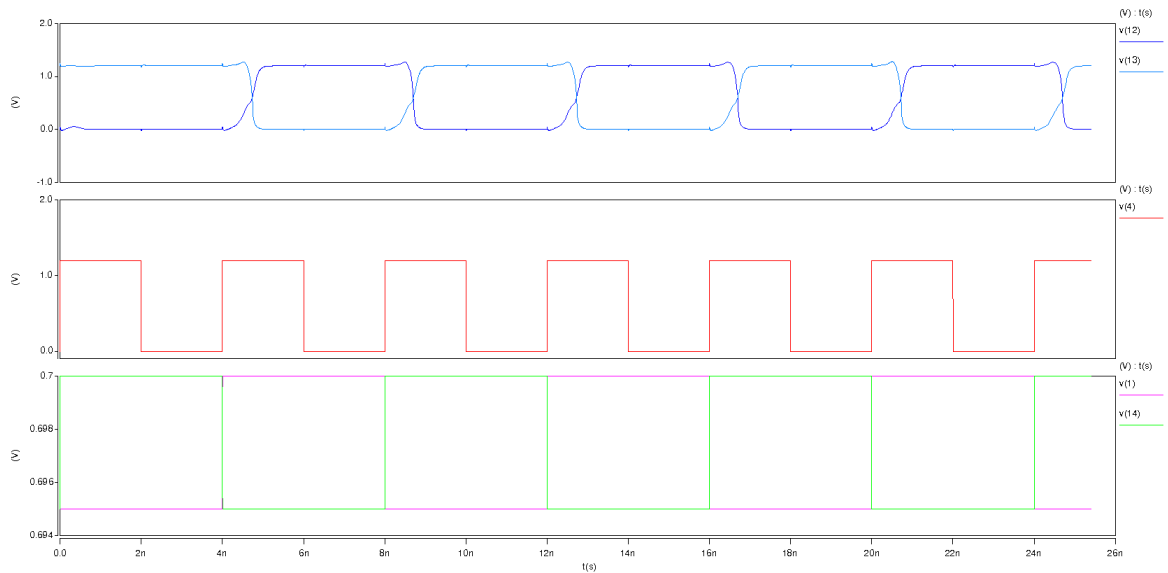
Figure 3.5. Transient simulation of the first solution.

4.   MULTI OBJECTIVE OPTIMIZATION

Providing flexibility for the designer to determine a particular solution among many candidate solutions is the main idea behind the MO optimization. For that purpose, all design objectives are required to be independent and equally important, which implies that they are not combined into a single function. Thereby, MO optimization allows interaction among many design objectives and provides a Pareto-Front (PF) including all feasible solutions. However, as discussed in the previous chapter, MO algorithms lose the efficiency when the number of design objectives are 3 or more. Thus, two objectives are given to the tool as design objectives and the rest are given as design constraints in order to be able to benefit from MO algorithms for the analog circuit synthesis tools.

4.1.   Algorithm Implementation

The Strength Pareto Evolutionary Algorithm (SPEA) is a popular technique for finding the Pareto-optimal set for MO optimization problems. The Strength Pareto Evolutionary Algorithm-2 (SPEA2), which is an improved version of SPEA, is chosen for implementation of the search engine as described in [25]. Although its superiority over the other multi objective methods such as Non-dominated Sorting Genetic Algorithm (NSGA2) and Pareto Envelope Based Selection Algorithm (PESA) is discussed in [9, 26], experience shows that efficiency of algorithms depend on the test scenario.

In Figure 5.8, the overall flow diagram of the algorithm is visually depicted. A SPICE netlist file needs to be prepared before beginning the optimization similar to explained in the SO algorithm. The superiorities of being topology and technology independent in addition to the advantages of being simulation based optimization are viable for the MO algorithms. The optimization starts with the initialization of an empty and a fixed sized archive and a fixed sized initial population, where the Sobol sequence is utilized again since it offers more homogeneously distributed initialization than the pseudo-random samples [27]. An additional population, called archive, is
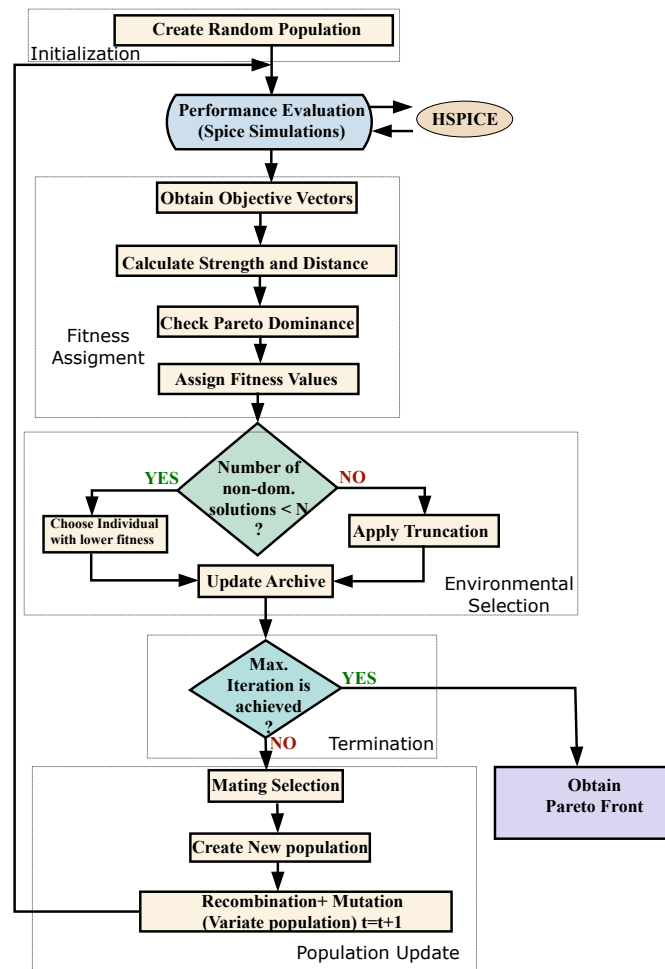
Figure 4.1. Flow diagram of the multi objective optimization.

created such that high performance members in the population are copied at each iteration. Although, the archive can be used as an external storage separate from the optimization engine, it is integrated into the algorithm since archive members are utilized in the selection process. In the next step, performance evaluation is performed for each individual in the population by performing SPICE simulations where HSPICE is used again. Then, a fitness value is assigned to each individual. It is important to emphasise that, fitness is different than the cost that defined in the SO optimization since it is not some combination of the design objectives. On the contrary, this value is calculated according to the density estimation and the Pareto dominance of each individual. The Pareto dominance is determined by comparing each design objective corresponding to an individual with others in both population and archive. The dom-

inance of the individual is determined if and only if all components of the individual are not smaller and at least one of them is greater than its competitor. Thus, all design objectives are treated independently, which enables to cover all range of possible solutions. By using Pareto dominance approach, each individual in the population and the archive is assigned to a strength value indicating the number of solutions it dominates, which is calculated as

$$S(i) = |\{j|j \in P_t + A_t \bigwedge i \succ j\}|. \tag{4.1}$$

The raw fitness that represents the sum of the strengths of its dominators in both population and archive is calculated as

$$R(i) = \sum_{j \in P_t + A_t \bigwedge j \succ i} S(j). \tag{4.2}$$

Also, density estimation is included into the raw fitness in order to separate the dominated individuals from the same individuals or having the same raw fitness value ones. To estimate the density, nearest neighbour technique is utilized, where the distance between individuals in the population and archive is calculated and stored in an increased order list. The kth element of the list determines the distance σ_i^k, where k is the square root of the sum of the individuals in the population and archive. Finally, the density and the fitness is calculated as

$$
\begin{aligned}
D(i) &= \frac{1}{\sigma_i^k + 2}, \\
F(i) &= D(i) + R(i).
\end{aligned}
\tag{4.3}
$$

Reducing the distance to the optimal front and increasing the diversity of population are major concerns for finding the Pareto-optimal set. To overcome these concerns, mating selection and environmental selection are proposed. Deciding on which individuals to keep during the optimization is called environmental selection. Therefore, the main goal of the environmental selection is to update the archive with respect to fitness values of individuals. First, all non-dominated solutions are stored in a list with respect to their fitness values. If the number of non-dominated solutions is less than the designer defined archive size, individuals with lower fitness values are chosen to fill the archive. If the size of the list exceeds the size of the archive, a truncation method, in which the individual with the smallest distance is chosen for removal, is performed iteratively until the pre-determined size is maintained. Then, the Pareto front is obtained if the maximum iteration number is achieved. Otherwise, mating selection is carried out, where a binary tournament method is utilized for random sampling of the union of the archive and population. In the tournament, two individuals from the archive are arbitrarily selected and the individual having the lower fitness is copied to the mating pool until it is filled. This approach is called the elitist selection, since only better individuals (those having lower fitness) are chosen as parents. It may cause the population to lose diversity or freezing; however, this situation is not observed since the maximum iteration count is less than the that of the SO optimization. Finally, recombination and mutation operators are employed to produce offspring and new population similar to explained in the previous chapter.

4.2. Synthesis Example

To test the developed tool, a two stage OTA circuit shown in Figure 3.2 was chosen as a design example, where the number of design parameters is 22, consisting of transistor dimensions, capacitors, resistors, and bias voltage. Electrical simulations were carried out with HSPICE by using the 130nm UMC technology and $1.2V$ supply voltage. The bandwidth and gain were selected as design objectives and the phase margin and the area were selected as design constraints. The population and archive sizes were determined as 50 each. The maximum iteration number for the termination

criterion was set to 20. The optimization process takes $20-30$ minutes, with an Intel i7 chipset with $2.80GHz$ processor. At the end of the optimization, the Pareto Front between bandwidth and gain is obtained as shown in Figure 4.2.
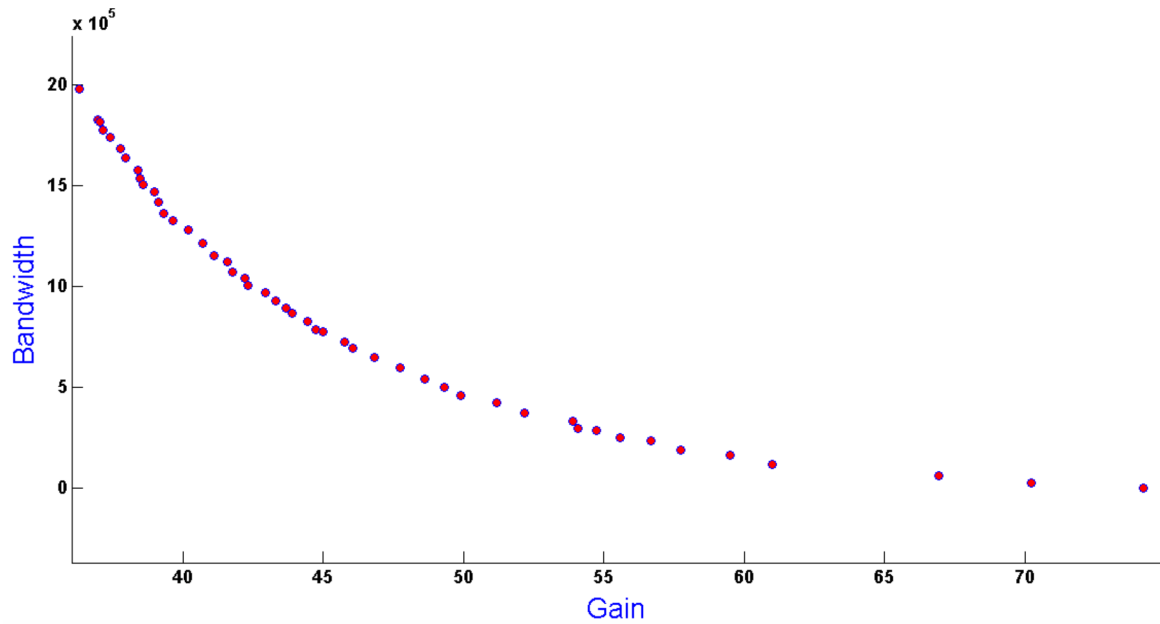
Figure 4.2. Pareto Front between bandwidth and gain.

5. YIELD-AWARE DESIGN AUTOMATION TOOLS

The aggressive downscaling of device geometries results in deterioration of process tolerances along with the device sizes. Therefore, it is a challenging problem to deal with process variations during fabrication process, which cause worsening reliability issues in CMOS circuits. For example, it becomes a headache to handle variations in the fabrication steps, such as line-edge roughness that is induced by gate etching and the lithography process [28], oxide thickness fluctuations that cause the fluctuation of the voltage drop across the oxide layer, affecting V_{th} , and random dopant fluctuations that significantly alter V_{th} [29]. Thus, circuit design without taking into account of variation leads to difference between the expected and the actual performance as shown in Figure 5.1. To prevent this discrepancy, some additional steps should be included in the conventional design procedure and analog design automation algorithms to achieve a certain performance after fabrication process [19,30]. As a result, in order to guarantee a certain yield for the design, some variability analyses to estimate the yield are required.
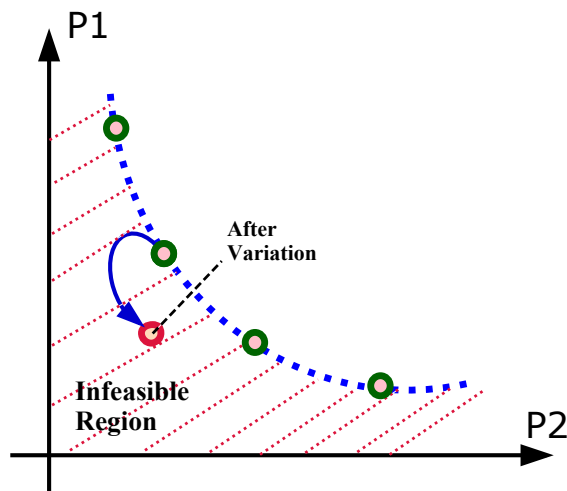
Figure 5.1. Optimum point moves towards to Infeasible region after variation.

5.1. Background

In the literature, sensitivity analysis [5], corner analysis [31], regression based models [32], and Monte Carlo (MC) based analysis have been commonly used for variability analysis [7, 19, 30]. Among them, Monte Carlo analysis is the most popular way to estimate the yield of a design, which is simply based on simulating randomly selected points in the uncertain parameter space and observing the variation effects on the yield. MC based approaches for estimating the effects of variations are prominently the most accurate [33]. However, the conventional MC approach requires a large number of simulations to provide a certain accuracy, which increases the computational effort. Considering the number of iterative evaluation simulations during optimization, it is inefficient to use the conventional MC technique for yield-aware circuit synthesis. To reduce the efficiency problem, several speed-up techniques have been developed in the literature. The idea behind these techniques is to minimize the number of samples by using either variance reduction techniques such as "Importance Sampling"(IS) and Latin Hypercube Sampling (LHS) or using Quasi Monte Carlo (QMC), that utilizes Low Discrepancy Sequences (LDS). Among these techniques, QMC has been the most efficient approach in terms of computational effort or CPU memory. The main advantage of QMC is scattering the samples on the space homogeneously rather than randomly. The other technique for efficient sampling is LHS, which has an excellent uniformity for one dimensional projections. However, QMC provides a superior uniformity for multi-dimensional projections [33]. One more important advantage of QMC exhibits itself for applications that require iterative sampling, such as yield-aware optimization. Since QMC is a deterministic approach, the sample size can be increased iteratively by pre-determined sample steps [19]. This feature is highly crucial during the optimization process to enhance the efficiency. Using the QMC approach for yield estimation, which promises adaptive sample size determination and automated stopping criterion mechanism, results in keeping the sample size to a minimum to avoid the redundant simulations.

5.2. Efficient Yield Estimation Techniques

There is a trade-off between the yield estimation accuracy and computational cost. MC based techniques are reliable and independent of the problem dimensionality, which makes them popular for yield estimation. The efficiency problem can also be improved by introducing some speed enhancement techniques. Conventional MC approach is based on random sampling of the uncertain parameter space [33]. However, random sampling causes clusters and empty spaces in their distribution over the sampling region and therefore requires a large number of samples to be able to spread out in the space. Considering the optimization process, there are many candidate individuals, for which yield analysis will be carried out, thus increasing the synthesis time dramatically. To reduce the synthesis time, one solution can be Infeasible Solution Elimination method (ISE), which is based on performing yield analysis only for the candidates that satisfy the user defined specifications [7]. On the yield estimation side, the efficiency of MC based techniques can be enhanced by using LDS. The main idea behind such approaches is to spread out the samples as homogeneously as possible to cover the whole design space with minimum a number of samples.

Conventional MC approach has an estimation error rate of $O(n^{-0.5})$ [33]. This error can be separated into the factor related to the function itself and the factor related to the generated set of random points according to the Koksma-Hlawka theorem [34], where the error is given as

$$\mid \widehat{Y} - Y \mid \le D_n^\star(x_1, x_2, x_3, ...x_n) V_{HK}(f). \tag{5.1}$$

In Equation 5.1, $\widehat{Y}$ and Y are the estimated and real values of the yield, respectively, and $D^\star$ is Star Discrepancy; measuring the uniformity of the generated points, where uniform distributions provide a smaller $D^\star$. $V_{HK}(f)$ is the total variance of the underlying integrand in the yield formula. As can be seen from this formula, the esti-

mation error can be decreased via two methods: increasing the uniformity of samples or decreasing the variance of the function f.

A common variance reduction method is the Latin Hypercube Sampling (LHS) method based on stratification [14], in which the term $V_{HK}(f)$ is reduced. The variance for one dimensional projections is highly reduced via LHS sampling, as reported in [14]. However, for higher order projections, the behaviour is similar to conventional MC [7]. A more efficient sampling technique is the Quasi-MC technique, which is based on lowering $D^\star$ via homogeneously generated samples. According to the Koksma-Hlawka theorem, homogeneous sample sets correspond to having lower discrepancy, which reduce MC estimation errors. The discrepancy of the conventional MC for n samples is given in [34] as

$$D^\star_{n|MC} = O\left(n^{-0.5}log(log(n))^{-0.5}\right), \qquad (5.2)$$

where the estimation error of the conventional MC is $O(n^{-0.5})$. On the other hand, considering QMC, the discrepancy is given in [7] as

$$D^\star_{n|QMC} = O\left(n^{-1}(log^s(n))\right). \qquad (5.3)$$

Low Discrepancy sequences provide an asymptotic integration error rate, which is much faster than the error rate of conventional MC method. In Figure 5.2, the star discrepancies of Conventional MC, LHS, and QMC are visually illustrated for 100 sample points and it can be observed that QMC has the lowest $D^\star$.

Several LDS strategies for generating sequences have been proposed such as Halton, Sobol, and Faure [35]. All of these LDS based strategies are deterministic, contrary to the random sampling performed in the conventional MC and LHS approaches. This deterministic behaviour becomes a superiority when an iterative variability analysis
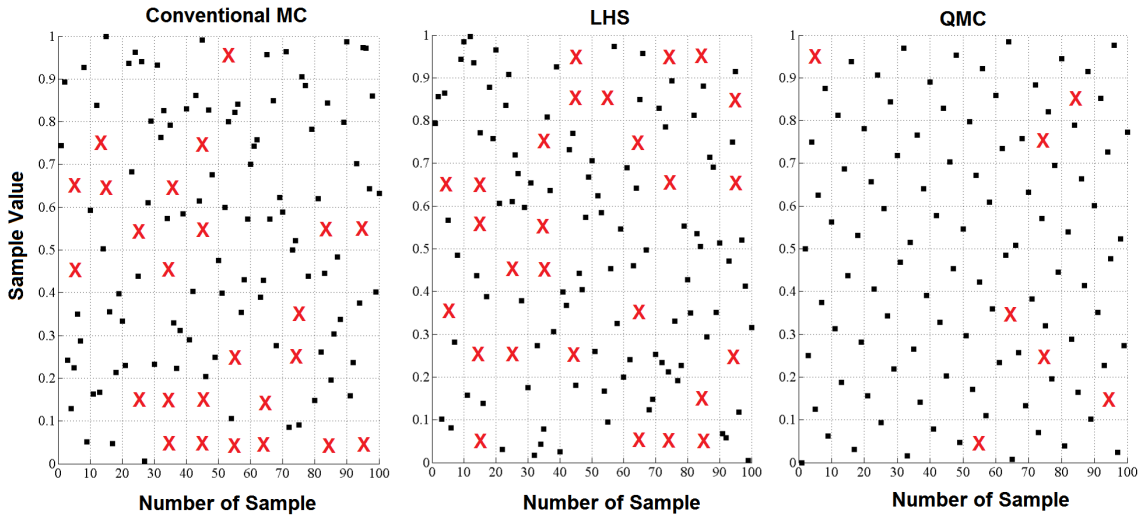
Figure 5.2. Distribution of samples generated by the Conventional, LHS, and Quasi MC approaches, where cross marks indicate empty region.

is required. Variability analysis is carried out many times during the optimization process and this results in longer synthesis times. Using constant sample sizes during yield estimation can still be problematic: keeping the size too small may lead to non-accurate estimations, whereas over-sampling may cause inefficiency in terms of CPU time.

5.3. Adaptive Sized Yield-Aware Analog Design Automation Using Single Objective Optimization

Considering variation effects, optimizing only electrical parameters is not sufficient, but also a certain yield for proper operation should be satisfied. As a result, the scope of analog design automation tools has been expanded to meet the yield requirement by including the yield constraint into the conventional objective minimization problem. The main idea behind yield-aware optimization is to find a dedicated solution region, where not only the electrical specifications, but also the yield requirement is satisfied. To address these problems, the tool proposed in [19] is utilized. The general flow diagram of this optimization tool is shown in Figure 5.3.

Figure 5.3. Flow diagram of the adaptive sized yield-aware tool.

5.3.1. User Interface and Optimizer

The tool was developed on the MATLAB® platform, where HSPICE was utilized for performance estimation. In the user interface module, circuit netlist, electrical

specifications, yield requirement, and trade-off coefficients for yield estimation are defined by the user. The tool described in Chapter 3 is utilized as an optimizer, where the pseudo code of the optimization algorithm is given in Figure 5.4, where g is the generation num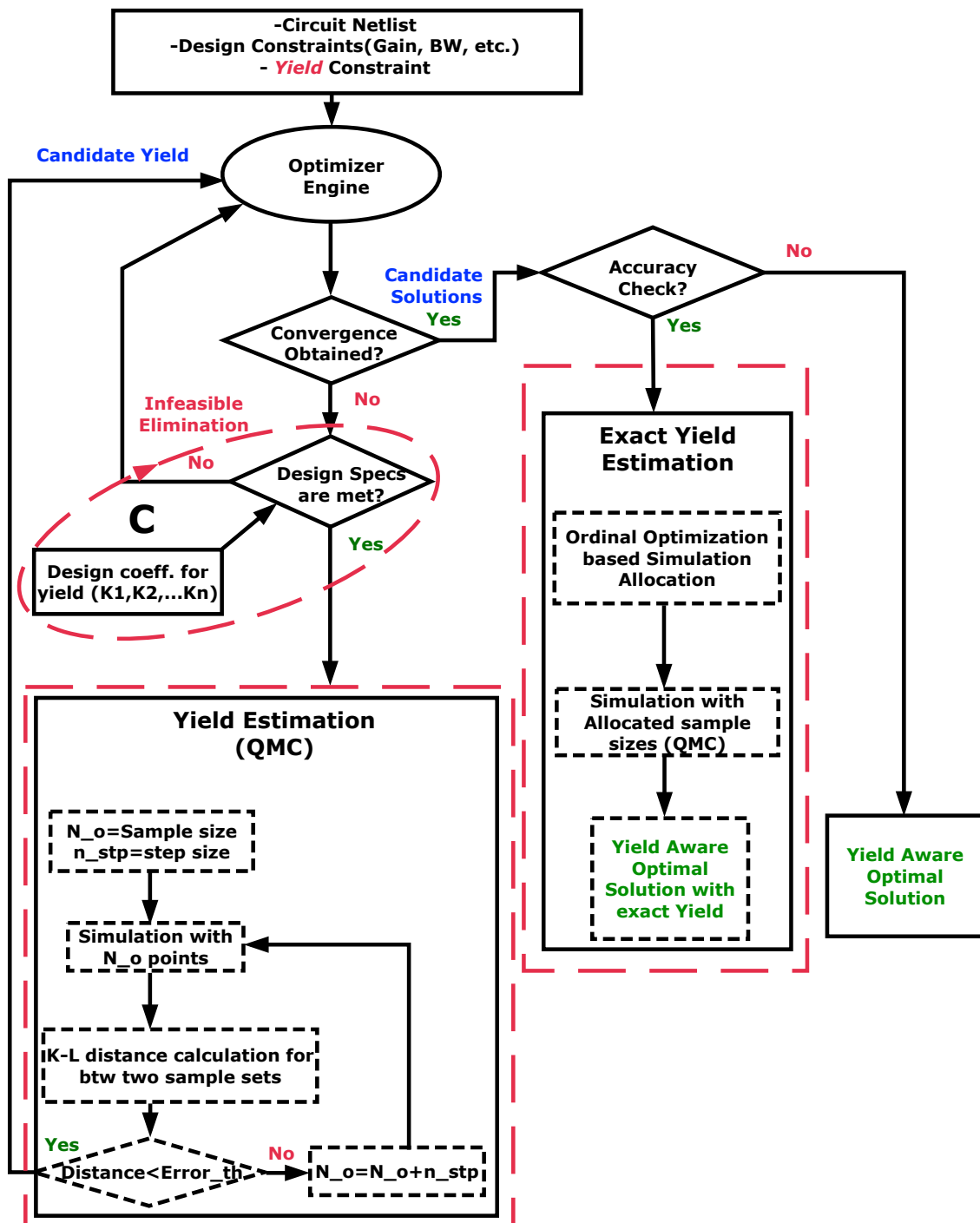ber, T is the population temperature, P is the current population, and μ and λ are the numbers of parents and offspring, respectively.

5.3.2. Infeasible Solution Elimination

The optimizer generally needs a few hundred generations of 100 individuals or a fraction of that for generations of 200 individuals. The total number of individuals remains more or less constant and reaches several thousands. Considering that at least a few hundred variability simulations are needed for yield estimation even for one candidate, the synthesis takes excessively long times. In order to decrease the synthesis time, the most common method is Infeasible Solution Elimination. According to this approach, variability analysis is only performed for the candidates that satisfy the performance metrics given by the user. As a result, yield analysis is only performed for solutions in the acceptance region; hence, the optimizer does not perform redundant simulations for infeasible solutions [7, 19].

5.3.3. Yield Estimation and Stopping Criterion

Solutions satisfying the feasibility test are selected as candidates and sent to the yield analysis module. QMC based variability analysis is performed for each candidate. A Sobol sequence is chosen to generate sample sets. The deviations in V_{th}, t_{ox}, W, and L are considered to evaluate the variation effects. The well-known Pelgrom Model [36] is utilized to calculate the variance and standard deviations of each device parameter.

To balance the trade-off between the efficiency and the accuracy during optimization, an adaptive sample size technique is required, which can easily be performed by QMC. Thanks to the deterministic generation of QMC samples, the sample size can be expanded within certain steps regarding the error rate between two neighboring sample sets. When the error rate considerably diminishes, the sample sequence expansion

```
g  ⇐  0
P_μ  ⇐  P_{μ0}
while  convergence  not  reached  do
    for  i= 1  to  λ/2  do
        [I_{parent1}, I_{parent2}] ⇐ choose(P_μ, 2)
        [I_{μ+i}, I_{μ+i+1}]_x ⇐ recombine_x(I_{parent1}, I_{parent2})
        [I_{μ+i}, I_{μ+i+1}]_s ⇐ recombine_s(I_{parent1}, I_{parent2})
    end


    for  i= 1  to  μ + λ  do
            if  I_i  is  selected  for  mutation
                    I_{i_x} ⇐ mutuate_x(I_i)
                    I_{i_s} ⇐ mutuate_s(I_i)
            end
        I_{icost} ⇐ evaluate(I_i)
    end
    P_μ^{g+1} ⇐ select(P_{μ+λ}^g, μ)
    g ⇐ g + 1
    T ⇐ update temperature()
end
    output ⇐ best solution
```

Figure 5.4. Pseudo code of the optimization algorithm.

is stopped. This point is the threshold point that is shown in Figure 5.5.
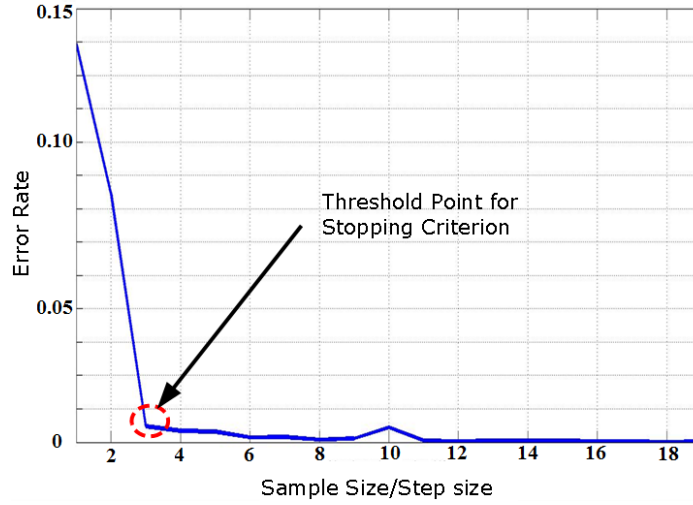
Figure 5.5. The concept of stopping criterion for adaptive sized QMC.

As mentioned, a self-adaptation mechanism was developed to determine the optimum adaptive sample size required by QMC. The Stopping Criterion is defined by the Kullback-Leibler (K-L) distance between histograms of two consecutive sample sets reducing to a pre-determined threshold value, which is depicted in Figure 5.5. The K-L [37] distance is a measure of the distance of two probability distributions over the same event space. It is the most frequently used information theoretic distance measure, which quantifies the similarity between two different probability distribution functions. The K-L distance of probability distributions P and Q on a finite set X is defined as

$$D_{KL}(P\|Q) = \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)}, \qquad (5.4)$$

where D_{KL} denotes the distance between two sample consecutive sets for our case. As can be seen from Figure 5.3, the sample size starts at a defined point, N_0, and is increased automatically by step size until D_{KL} decreases down to the threshold point.

After the variability analysis part, the candidate solutions are transmitted with their varied outputs to the yield calculation part. In this part, yield values for each

candidate are calculated by sampling the uncertain parameter space several times and performing variability simulations for each sample set. Variation effects on the circuit performances are obtained throughout simulations and finally, yield is estimated by using a user defined specific region, which is called as "Acceptance Region", and determined according to

$$A = \{x : y_m \geq K_m.O_m|, \ x \in \mathbb{R}^s\}, \tag{5.5}$$

$$I_A = \begin{cases} 1, & x \in A \\ 0, & x \notin A, \end{cases}$$

where m is the number of outputs, $K_1, K_2, ..., K_m$ are trade-off coefficients defined by the designer and O denotes circuit outputs. I_A is the characteristic function of the acceptance region, also known as indicator function [33]. Yield is calculated whether the outputs exist in the acceptance region or not for each sample set. The expected value of the indicator function is also used for yield calculation, which is defined as

$$\begin{aligned} Y &= P(x \in A) = E(I_A(x)), \\ Y &= \int_{\mathbb{R}^s} I_A(x)\pi(x)dx, \end{aligned} \tag{5.6}$$

where $\pi(x)$ is marginal distribution of parameter x. This s-dimensional integration takes a canonical form as

$$Y = \int_{C^s} f(x)dx, \tag{5.7}$$

where C is the unit cube in s dimensions. Numerical approximation of this integral is expressed as

$$Y_N = \frac{1}{N} \sum_{i=1}^{N} f(x_i),$$ (5.8)

where x_i denotes sampled points.

At the end of this part, candidate yield is sent to the optimizer engine and is included to the optimization process by calculation of the contribution to the cost. The total cost is updated with respect to the estimated yield, thus the yield is evaluated as a design constraint as well as electrical constraints (e.g gain, bandwidth, power, etc.).

5.3.4. Exact Yield Estimation/Simulation Budget Allocation

This module is an optional part of the tool and will be active if the user chooses so. The idea behind this module is to allocate a pre-defined simulation budget, T_{budget}, which is determined as given in (Equation 5.9), to each candidate solution based on its yield value to observe a more accurate yield estimation. To determine the simulation amount for each candidate, an asymptotic approach proposed in [38, 39] is as

$$n_b = \sigma_b \left(\sum_{i=1, \ i \neq b}^{M_1} \left(\frac{n_i^2}{\sigma_i^2} \right) \right)^{0.5},$$

$$\frac{n_i}{n_j} = \left(\frac{\sigma_i / \delta_{b,i}}{\sigma_j / \delta_{b,j}} \right)^2,$$ (5.9)

$$i, j \in 1, 2, 3, \dots N_{cand}, \ i \neq j \neq b,$$

$$T_{budget} = sim_{ave}.N_{cand},$$

where b is the best candidate according to the yield specifications among N_{cand} candidates. $\sigma_1^2, \sigma_2^2, \sigma_3^2, \dots, \sigma_{N_{cand}}^2$ denote the variance values of N_{cand} solutions. $\delta_{b,i}$ denotes the deviation of the estimated yield with respect to the yield of the best design and sim_{ave}

is a user-defined variable arranging the average value of the number of simulations per candidate. The algorithm optimizes the trade-off between deviations of estimated yield and accuracy to obtain reliable yield estimation.

5.3.5. Synthesis Examples

Two stage OTA and folded cascode amplifier were chosen as design examples, which are shown in Figures 3.2 and 3.3 respectively. All circuit elements (transistor dimensions, bias voltages-currents, etc.) were given to the optimizer as design parameters, which is 22 for two stage OTA and 16 for folded cascode amplifier. Simulations were performed on HSPICE using 130nm UMC technology. ΔV_{th}, Δt_{ox}, ΔW, and ΔL were considered during the yield estimation and the mismatch model provided by UMC 130nm was used during variability analysis. The standard optimization process takes $15 - 20$ minutes whereas yield-aware optimization takes $60 - 70$ minutes with an Intel i7 chipset with 2.80GHz processor. Numbers of parents and offspring, and maximum number of iterations were chosen as 200, 200, and 20, respectively. The initial value for sample size (N_0) and the step size (n_{stp}) were chosen as 48 and 10, respectively. To determine the total simulation budget for exact yield estimation, sim_{ave} was selected to be 100. Synthesis results of each circuit topology are presented in Table 5.1 for 3 independent runs for both standard and yield-aware optimization. Constraint coefficients, $K_1, K_2, ... K_5$, for infeasible solution elimination were selected as 0.9. According to the sample size results, $98 - 128$ samples were used to estimate the yield during the synthesis. After the synthesis process, simulation budget allocation for exact yield estimation was enabled, and as seen from the table, the maximum number of additional samples was devoted to the candidate with maximum yield. This number is an order of magnitude larger than the sample sizes used during the optimization process. The maximum error rate in the yield is found to be 2.96%, which indicates that the stopping criterion is quite successful in determining the minimum sample size required for accurate yield estimation. As expected, the chip area and power consumption increase to satisfy the yield constraint. Comparing to the standard optimization results given in Table 3.1, elecrical specifications are poorer. This is due to the fact that yield is

taken into account as a design constraint contrary to the optimizer, which focuses only on the electrical specifications.

Table 5.1. Synthesis results of yield-aware optimization for 3 independent runs.

	BW(kHz) >15kHz	Gain(dB) >55dB	PM(^{o}) >60°	Power(μW) <1.5μW	Area($e^{-9}m^2$) <1$e^{-9}m^2$	Candidate Yield >70%	Sample Size	Exact Yield(%)	Extra Allocated Samples	Error (%)
						Two Stage OTA Results				
1	18.2	56	88	4.2	1.9	87.96	108	90.92	1081	2.96
2	26	55.6	73	2.1	3.1	94.83	98	95.22	1552	0.39
3	20	28.5	76	1.5	2.3	100	128	98.84	21034	1.16
						Folded Cascode Results				
	BW(kHz) >20kHz	Gain(dB) >55dB	PM(^{o}) >60°	Power(μW) <1.5μW	Area($e^{-9}m^2$) <1$e^{-9}m^2$	Candidate Yield >70%	Sample Size	Exact Yield(%)	Extra Allocated Samples	Error (%)
1	42.5	56	58	1.8	2.86	76.7	128	78.24	617	1.52
2	19.5	52	67	2.8	2.75	91.8	108	91.46	2985	0.36
3	46.5	50	69	2.1	3.96	100	108	97.46	5049	2.54

5.4. A Hybrid Method for Yield-Aware Design Automation Tools

The major drawback of the QMC approach is that there is no practical way to know the error in the estimated yield, since it is impossible to calculate the total variation $V(f)$ in Koksma-Hlawka inequality. As a result, confidence interval of the estimation can not be obtained. To overcome this issue, LDSs are randomly permuted by scrambling [30, 40], which simply reorders the sequence of values independently. Therefore, a few differently scrambled QMC runs provide a standard deviation that can be used as a probabilistic measure of the estimation error [30, 41]. However, the requirement of multiple runs may degrade the synthesis time. The proposed approach described in [30], promises a solution to estimate the error bounds of estimated yield while sustaining the time efficiency of the optimizer, in which QMC and scrambled QMC are conducted together.

5.4.1. Yield Estimation Method

In previous section, QMC approach is used during the yield estimation part of the developed analog optimization tool, and it is demonstrated that a few hundred LDS points are sufficient to make quite accurate estimation. On the other hand, the disadvantage of the QMC approach is that the statistical error in the estimation cannot

be calculated since deterministic samples which have no natural variance are utilized. Also, in higher dimensions, it is hard to calculate the exact values of the error $(Y - Y_N)$, because variance of integrand V(f) becomes intractable. Even if V(f) is calculated and an upper bound is obtained, the estimated and exact value of integration would be very different. To overcome this bottleneck, and obtain a confidence interval of estimated yield, scrambled-QMC technique [40], which is based on permuting the order of the sample set within a random manner, is exploited to obtain artificial statistical variance, as described in [30, 42]. A scrambled-QMC sample set, $\{x_i^{(j)}\}_i^N$, $j = 1, 2.., M$, is selected and the yield is estimated for each sample set as

$$Y^{(j)} = \frac{1}{N} \sum_{i=1}^{N} f(x_i^{(j)}), \quad j = 1, 2..., M. \tag{5.10}$$

Then the mean of the yield is calculated as

$$\widehat{Y} = \frac{1}{M} \sum_{j=1}^{M} Y^{(j)}. \tag{5.11}$$

The error of numerical integration is estimated using the variance of the evaluated yield values, which is calculated as

$$\widehat{\sigma}^2 = \frac{1}{M(M-1)} \sum_{j=1}^{M} (Y^{(j)} - \widehat{Y})^2. \tag{5.12}$$

Finally, the magnitude of the QMC error is calculated as

$$|E_{QMC}| = \widehat{\sigma}.\Phi^{-1}(\frac{1+p}{2}) \tag{5.13}$$

with user defined probability p, where Φ is the standard normal cumulative function. As a result, thanks to the randomness property of scrambled QMC, the minimum and the maximum bounds of yield with probability p can be obtained. To utilize the superiorities of each method, a hybrid solution is implemented and embedded into the yield estimation block shown in Figure 5.6.
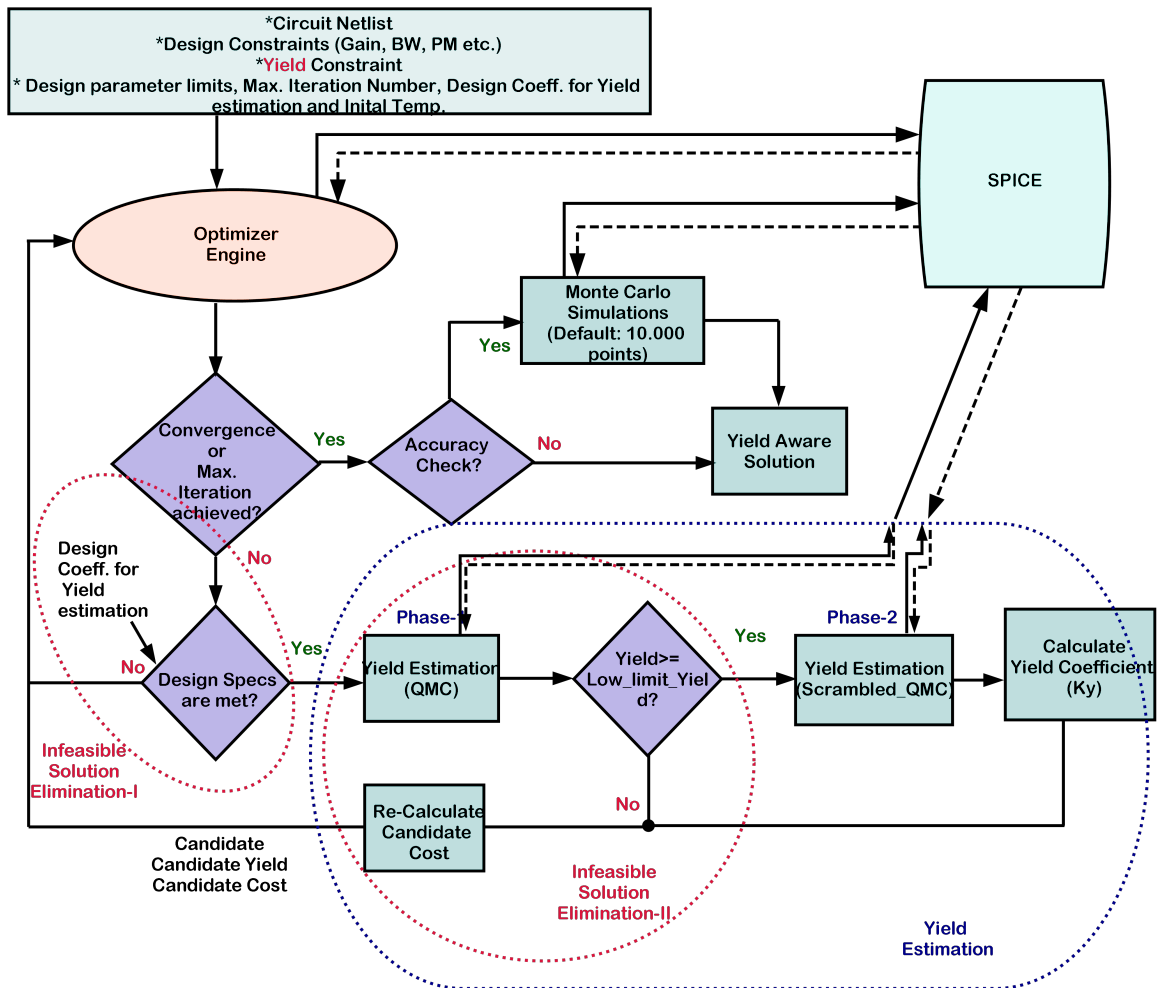

Figure 5.6. Flow diagram of the algorithm.

5.4.2. Algorithm Implementation

During the optimization process, each individual in the population is evaluated to determine whether it is inside or outside of acceptance region. The imprecise specifications for analog design optimization may lead to find a better solution than deterministic specifications [7]. For instance, slightly losing from one electrical constraint may cause much better performances for the other constraints, as a result enhancing the overall circuit performance. Thus, design coefficients are added to the developed tool in order to arrange trade-offs between each design constraint. Variation analysis is employed only to solutions in the acceptance region. Therefore, redundant simulations are avoided and synthesis time is reduced since variation analysis is not employed to infeasible solutions. On the other hand, if an individual satisfying the acceptance region is found, variation analysis and yield estimation are performed in two consecutive phases. In the first phase, a single QMC is run, in which Sobol sequence is preferred among other LDS sets since it is empirically shown that it provides better results in higher dimensions as stated in [43]. Also, the first N points can be skipped in order to achieve more homogeneity, and thus, better sampling performance [41]. In the second phase, a further analysis is employed to obtain standard deviation of the yield. Before the second phase of yield estimation, another infeasible solution elimination is performed considering the yield found in the first phase. If the yield value is less than the minimum required yield, the solution is eliminated and the second phase of yield analysis is not performed. Thus, the synthesis time is reduced since many redundant simulations are not run. When a solution survives the first phase and satisfies the second ISE, additional scrambled QMC based simulations are run to obtain standard deviation and magnitude of the error as given in Equation 5.13. At the end of this part, upper-lower bounds and standard deviation of the estimation are obtained. The additional cost factor is calculated using the worst case to guarantee a yield satisfying the minimum requirement, and then added to the objective cost. Here, a particular ranking is applied to the candidate and a new defined yield coefficient is calculated as

$$K_y = \frac{|Y_{best} - Y_{lower}| + \epsilon}{Y_{upper} - Y_{lower}} \tag{5.14}$$

where Y_{lower}, Y_{upper} and, Y_{best} denote user defined yield constraints and the maximum bound of instant yield. The term ϵ is added to avoid zero cost value, which was determined $\simeq 0.1$ for practical implementation. The solution having a high yield value is assigned to minimum coefficient. Then, the cost of the candidate solution is multiplied by the yield coefficient to reduce the cost of the solution with respect to its yield. This approach increases the probability of the selection of better individuals in terms of yield, thereby, the population is oriented to find more robust solutions throughout the evolution. When convergence or maximum iteration number is reached, a larger sized pseudo-random Monte Carlo is applied to the found solutions to provide precise estimation.

5.4.3. Synthesis Examples

In order to test the developed tool, the two stage OTA shown in Figure 3.2 and folded cascode amplifier shown in Figure 3.3 were chosen as design examples again. Deviations at V_{th}, t_{ox}, W, and L were considered in variation analysis during the yield estimation. The mismatch model provided by UMC 130nm was utilized to mimic process variation phenomenon. The standard optimization process takes $15 - 20$ minutes whereas yield-aware optimization takes $60 - 70$ minutes with an Intel i7 chipset with $2.80GHz$ processor. Numbers of parents and offspring, and maximum number of iterations were chosen 100, 100, and 50, respectively. The number of sample sizes for QMC and scrambled QMC was determined 100. In order to obtain error bounds of estimation, the number of scrambled QMC run was kept as small as possible ($M = 2$) considering the time efficiency. Although the higher number of runs provides more accurate standard deviation and mean of estimation, it is not necessary to run additional simulations as long as the approximate error bound is known. As a result, the standard deviation was calculated for 3 independent runs for each candidate. Con-

straint coefficients, $K_1, K_2, ... K_5$, for infeasible solution elimination are selected 0.9. The sample size of the final Monte Carlo analysis was determined as 10.000.

Synthesis results of each circuit topology for 3 independent runs are presented in Table 5.2. According to the table, both electrical constraints and yield are satisfied except the chip area and power consumption with trade-off to achieve the demanded yield. Moreover, Monte Carlo simulation results with 10.000 samples prove the accuracy of the proposed yield estimation part where all exact yield values are found in the confidence interval with a maximum error of 1.46%, which is determined during the optimization process. In Figure 5.7, Figure of Merit versus yield is plotted for a couple of solutions that was found by the optimizer in a single run for Folded Cascode example, in which there are some solution candidates having high yield, but lower FOM and vice versa. The final decision is determined by the designer considering the trade-off between FOM and yield constraints.
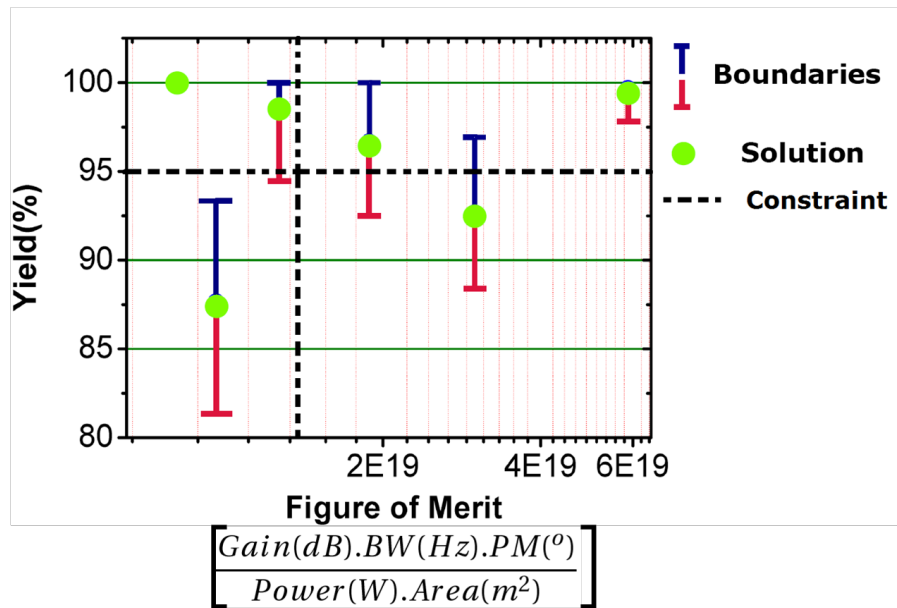

Figure 5.7. FOM vs Yield.

Table 5.2. Synthesis results of 3 independent runs.

	BW(kHz)	Gain(dB)	PM(^{o})	Power(μW)	Area($e^{-9}m^2$)	Yield(%)	Standard Deviation	Confidence Interval (99%)	Exact Yield (%)	Error (%)
Two Stage OTA Results										
	>10kHz	>55dB	>60°	<1.5μW	<1$e^{-9}m^2$	>80%				
1	21	58	77	2.3	2.62	99.33	1.15	$96.4 \leq x \leq 100$	98.84	0.49
2	10.2	62.2	68.3	2.5	3.57	93.10	4.72	$83.37 \leq x \leq 100$	95.70	1.46
3	53.7	55.2	65	2.6	3.5	98.4	0.57	$97.87 \leq x \leq 100$	97.94	0.47
Folded Cascode Results										
	BW(kHz) >20kHz	Gain(dB) >50dB	PM(^{o}) >55°	Power(μW) <1.5μW	Area(e^{-9}) <1e^{-9}	Candidate Yield(%) >80%	Standard Deviation	Confidence Interval (99%)	Exact Yield (%)	Error (%)
1	27	54	57	3	2.17	99.3	1.14	$96.41 \leq x \leq 100$	98.12	1.2
2	34	53	57.1	3	2.18	97.8	0.58	$95.87 \leq x \leq 98.8$	95.94	1.42
3	108	53	59	1.7	2.81	99.66	0.57	$98.2 \leq x \leq 100$	99.9	2.54

5.5. Yield-Aware Analog Design Automation using Multi Objective Optimization

To test the developed methodologies for yield estimation, MO optimization tool described in Chapter 4 is also utilized. The flow chart of the developed tool is given in Figure 5.8, where the multi-objective search engine and scrambled QMC yield estimation method are combined [25]. The yield is given as a new objective in addition to the electrical objectives. In contrast to the nominal optimization, where only electrical performance evaluation is employed via SPICE simulations, the yield analysis part is also included in the performance evaluation block to estimate the approximated yield values as given in [25]. The yield of each individual is estimated according to the measured electrical parameters as their nominal value. Then, fitnesses are assigned to individuals and environmental selection is applied to update the archive. Then, the mating selection method is utilized to select candidates, which are used to construct the new population by using crossover and mutation operators. This procedure is repeated until the termination criterion is met. Finally, a yield-aware Pareto front can be obtained by using the individuals in the archive at the end of optimization.

5.5.1. Synthesis Example

To test the developed tool, a two stage OTA circuit shown in Figure 3.2 was chosen as a design example, where the number of design parameters is 22, consisting
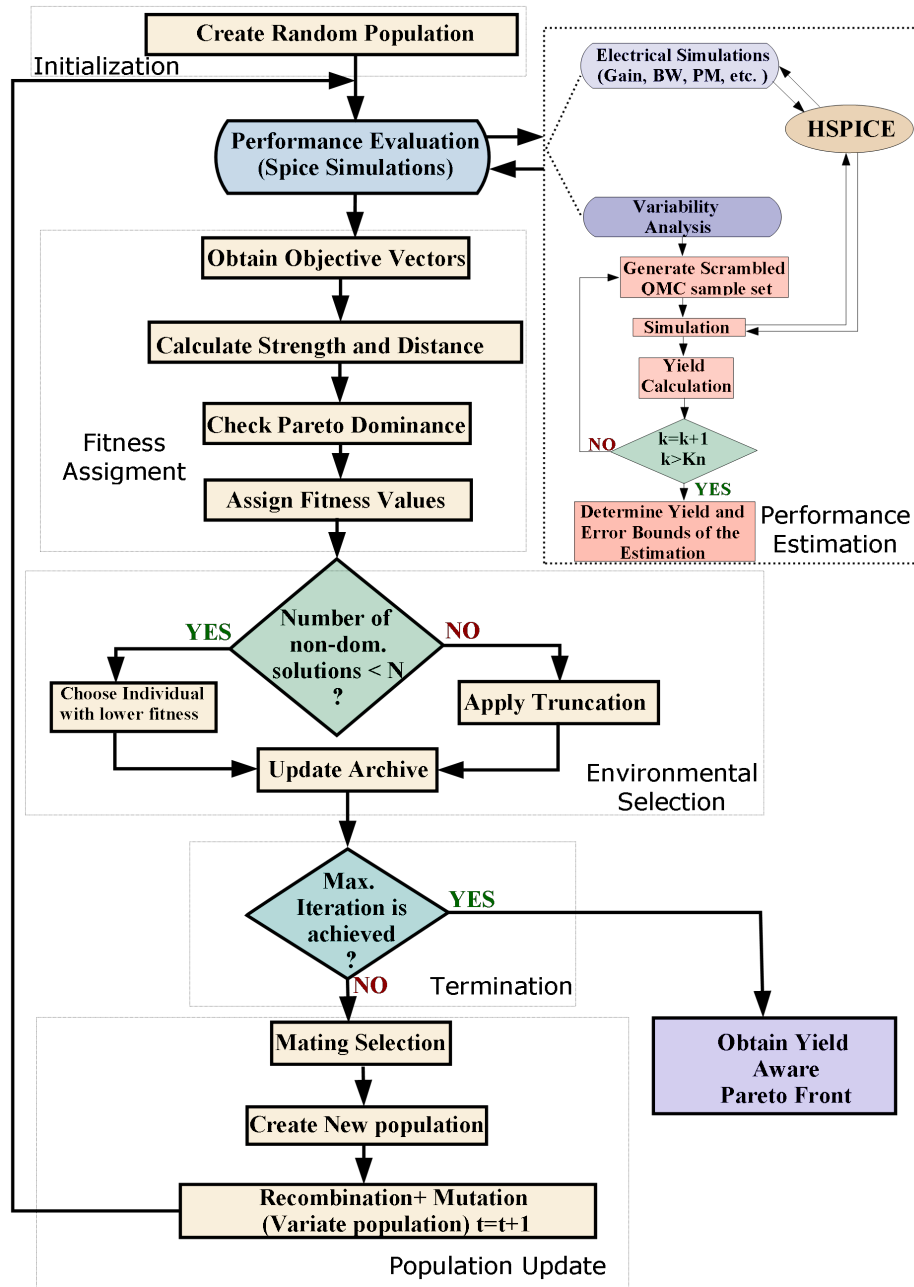
Figure 5.8. Flow diagram of the proposed tool.

of transistor dimensions, capacitors, resistors, and bias voltage. Electrical simulations were carried out on HSPICE utilizing the 130nm UMC technology. Variations in the transistor width, length, threshold voltage, and oxide thickness were considered to examine the process variation phenomenon. Pelgrom Model [36] was used to obtain probability distributions and standard deviations of uncertain parameters, where the

yield and circuit performances are observed by sampling these distributions with three different scrambled LDS sets, each consisting of 50 points. The main algorithm was implemented on MATLAB. The 3-dB cut-off frequency, gain, phase margin, and the yield were selected as design objectives. The population and archive sizes were determined as 50 each. The maximum iteration number for the termination criterion was set to 20. The nominal optimization process takes 20-30 minutes, whereas the yield-aware optimization takes almost 5 hours with an Intel i7 chipset with $2.80GHz$ processor. At the end of optimization, the Pareto front between the yield and the Figure of Merit (FOM) is given in Figure 5.9.
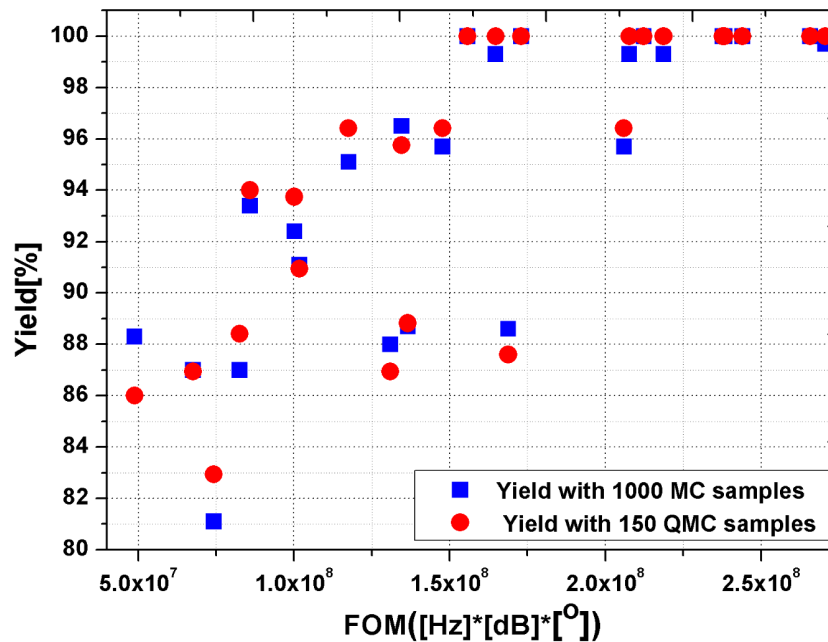

Figure 5.9. FOM vs Yield for multi objective optimization.

As seen from Figure 5.9, both the yield and electrical design objectives are optimized by introducing yield as a design objective, which allows the designer to access more robust solutions with different FOM values. Furthermore, in order to illustrate the accuracy of the yield estimation, MC analysis with 1000 random samples is performed for all individuals on the Pareto front. The maximum estimation error was calculated as 2.3%. In Figure 5.10, a couple of solutions on the Pareto front with different yield values are plotted to illustrate the effect of process variation. As seen

from Figure 5.10, solutions having lower yield are more prone to move out of the front.
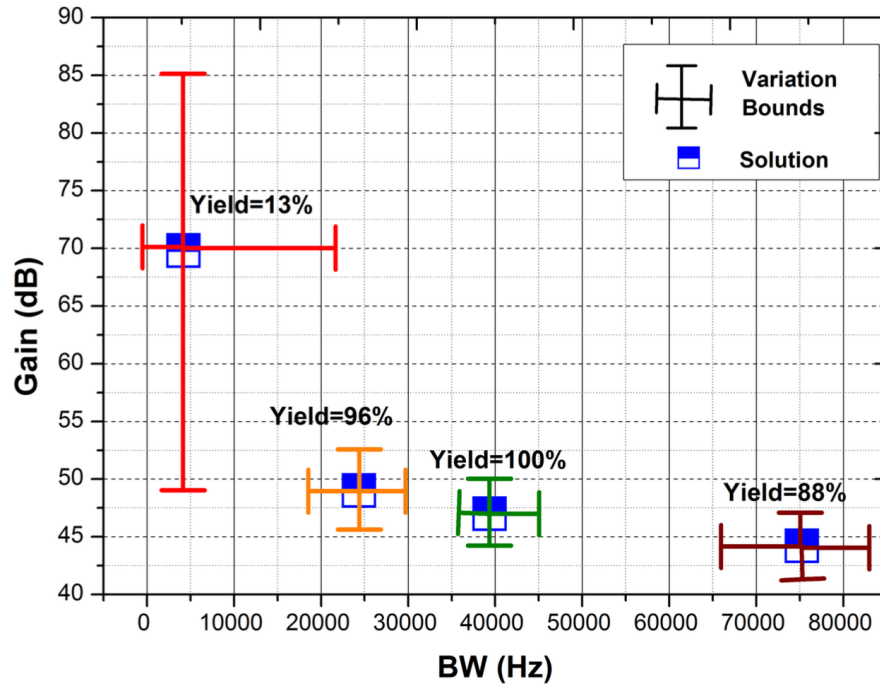

Figure 5.10. Effects of variations on the Pareto front.

6. LAYOUT-AWARE ANALOG DESIGN AUTOMATION TOOL

6.1. Background

There are circuit sizing and layout generation tools for analog circuit designers to speed up the design process. Generally, these tools handle the circuit sizing and the layout generation processes separately, which may cause performance failures and laborious redesign iterations [44]. Recently, new tools have been developed, which simultaneously take care of circuit sizing and layout generation. However, they either suffer from long run times or limited accuracy of the utilized parasitic model. In this thesis, a complete layout-aware design automation tool for analog circuits is presented. The proposed tool combines a simulation-based circuit sizing tool with a template-based layout generation tool [10, 44]. The layout-induced parasitics are automatically extracted via a commercially available extractor.

There are several CAD tools supporting layout-aware circuit sizing as given in [45–55]. The tools in [47, 49] use analytical models to estimate layout parasitics. In [19, 46], the overlapping, fringing, and routing parasitics for critical nets are extracted. In [55], the layout area is calculated from the generated floorplans. The tool in [51], estimates some of the layout parasitics. In [52], a layout retargeting tool is presented, where the parasitics are modeled. A layout-aware automatic sizing flow for retargeting analog circuits is proposed in [53], where the interconnect parasitics are estimated. A design methodology is presented in [54], where the design plans are embedded to compute parasitic effects. In [50], the parasitics are included in the performance models by symbolic analysis. The work in [48] follows a two-step approach, a global and then a detailed design. In the global design step, only a very rough initial solution is found, which does not contribute much to the elimination of infeasible solutions from the detailed design step. The work in [45] uses a template to generate a layout and minimizes the layout area also the parasitics are extracted in each iteration. In [56],

a layout is generated through optimization in every iteration, and the parasitics are extracted.

6.2. A Two-Step Layout-in-the-Loop Design Automation Tool

As the complexity of a circuit increases, circuit-level sizing and layout generation processes become more challenging. A typical manual design flow is given in Figure 6.1a. The flow starts with circuit sizing, which is conducted until all performance specifications are satisfied. It is followed by the layout generation, in which the layout components are placed as compact as possible and routed. If all of the performance specifications are not met after the post-layout simulations, the layout has to be modified to reduce the layout induced parasitics. In the worst case scenario, the whole process is repeated starting from the circuit sizing. Thus, the main disadvantage of this design procedure is that it requires highly experienced designers, otherwise it is very time consuming [44].
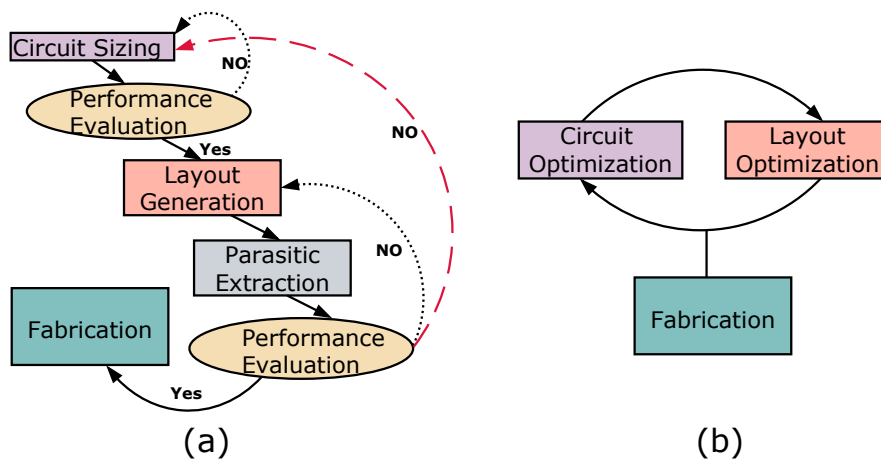

Figure 6.1. a) manual design flow b) proposed two step approach.

The developed tool proposes a two-step approach, demonstrated in Figure 6.1b. In the first step, a simulation-based circuit sizing optimization is performed that uses SPICE simulations without taking care of the layout induced routing parasitics. When all the performance goals are attained, a layout is generated and the parasitics are extracted for the solutions that achieve the performance specifications. In the case where

the solutions satisfy the circuit specifications in the presence of layout parasitics, a feasible solution is achieved and the loop ends. Otherwise, parasitic-aware optimization is performed, in which the solutions found in the first step are accepted as an initial population and a local search is conducted to compensate the performance loss. In this step, a layout is generated for every solution and its parasitics are precisely extracted via a commercially available extractor. The tool stops when all the performance goals are met in the existence of layout induced parasitics. The proposed approach, as described in [44], utilizes a precise extractor and the overall run time is acceptable due to elimination of time consuming post-layout simulations for infeasible solutions.

6.2.1. Circuit Optimization Tool

The overall algorithm is developed on the MATLAB® platform, where HSPICE was utilized for performance estimation. Circuit netlist and electrical specifications are defined by the user. The tool described in Chapter 3 is utilized as an optimizer, in which conceptual flow diagram of the circuit optimization tool is shown in Figure 6.2
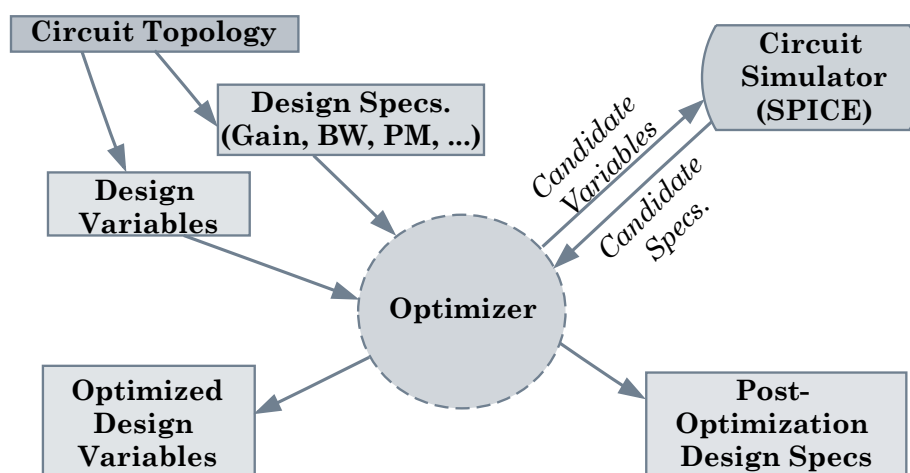

Figure 6.2. Conceptual flow diagram of the optimizer.

6.2.2. Layout Generation Tool

Layouts are instantiated from a template and the area of the layout is minimized by changing free geometric parameters, such as the number of fingers of a transistor. The template language, LDS [57], supporting both sequential and constraint programming, is used to code the template. The general methodology in [58] is used to minimize the layout area. The template contains the code for placement and routing, as well as an optimization loop to minimize the area. The input parameters, which specify device dimensions, are read from a file, and the tool outputs a GDS file that is validated via a commercial design rule checker. The block diagram in Figure 6.3 demonstrates the procedure of template instantiation. Device generators are called from a technology library, which was also coded in LDS. To improve matching, some of the transistors are interdigitized. The groups of interdigitized transistors are shown in Figure 6.4, and named as P, N1, N2, and N3. The widths and lengths of the transistors in a group must be equal, which is imposed as a constraint in the circuit synthesizer.
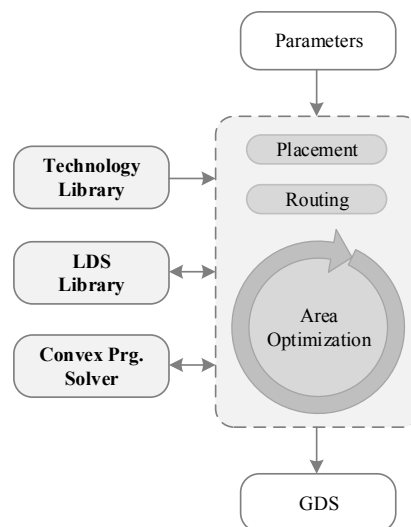

Figure 6.3. Flow of template instantiation.

A channel router is called from the technology library to make the internal routing of the interdigitized transistors. The routing between the interdigitized transistors
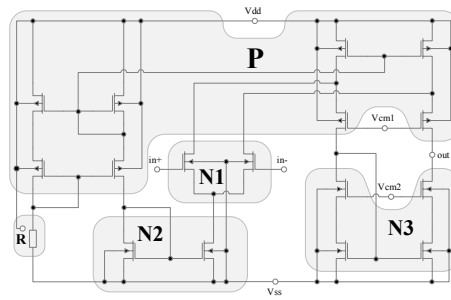
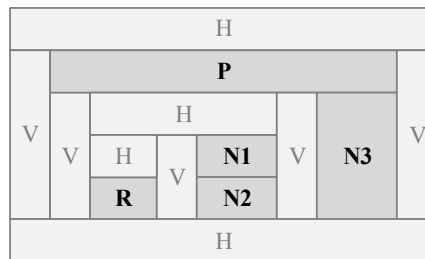Figure 6.4. Template of interdigitized transistors.

Figure 6.5. Floorplan for the circuit in Figure 6.4.

P, N1, N2, N3, and the resistor R are done via vertical and horizontal routing channels. The floorplan of the template is shown in Figure 6.5, where the letters H and V represent the horizontal and the vertical routing channels, respectively. The geometric parameters of the interdigitized transistors and the resistor, and the number of fingers are determined in the area optimization loop in Figure 6.3, where a convex programming problem is generated and solved via a commercial solver.

6.2.3. Synthesis Example

The folded cascode OTA shown in Figure 6.4 is used as an example analog circuit in order to test the developed tool. The transistor sizes and the bias resistance value are determined as design parameters. The number of parents, offspring, and maximum iteration number are chosen as 50, 50, and 200, respectively. Electrical design objectives are the gain, f_{3dB}, and phase margin where lower and upper bounds

are set to $55 - 60dB$, $9 - 10kHz$, $50 - 60^o$, respectively. However, solutions that achieve higher electrical design objectives than upper bounds are not penalized. The whole algorithm is coded in MATLAB, where electrical simulations are carried out using HSPICE and a 130nm CMOS technology is used. By using an Intel i7 chipset with $2.80GHz$ processor, circuit sizing without considering the parasitic effects takes $25 - 30$ minutes. At the end of the sizing step, design parameters of the solutions that satisfy all electrical performances are sent to the layout generation tool. The time to output a GDS file is a function of the number of design parameters of the solution, which is used to determine the number of fingers. The run time average over 100 runs is 0.3s, including the time required for the I/O operations and the time spent during the area optimization. At that point, the search space is narrowed and only small deviations in the design parameters are allowed. For each individual, a GDS file is generated and parasitics are extracted. Thus, parasitic-aware circuit sizing is achieved, in which both electrical objectives and parasitic effects are considered while obtaining the optimum physical layout. However, relatively large CPU time is spent since extracting the parasitic file at each iteration for every candidate takes a long time. To reduce this time, population and offspring sizes are reduced to 10 individuals. In addition, number of iterations for parasitic-aware sizing is determined as 3. Synthesis results are given in Table 6.1.

Table 6.1. Synthesis results.

Solutions	Optimization Results			Post-Layout Results			Parasitic Aware Optimization Results		
	Gain (dB)	f_{3dB} (kHz)	Phase Margin $(^o)$	Gain (dB)	f_{3dB} (kHz)	Phase Margin $(^o)$	Gain (dB)	f_{3dB} (kHz)	Phase Margin $(^o)$
1	57.48	33.43	72.19	58.71	63.76	57.88	-	-	-
2	58.9	37	68.08	51.1	125	68.11	55.67	54.17	58.05
3	58.16	40	69.55	53.88	137.4	71.78	56.4	86.37	60.74

The first column group presents the results of the circuit optimization without taking parasitics into account. The second column group shows the post layout results of the sample individual from the first column group. The third column group presents the results of the parasitic-aware optimization of the circuit. It can be observed that the electrical results of Solution 1 satisfy the design objectives, when parasitic effects are included. Thus, parasitic-aware sizing is not performed for this solution in order

to speed up the total CPU time. Also, it can be shown that, electrical parameters of Solution 2 and 3 are worsened after adding the parasitic effects. Thereby, parasitic-aware sizing is performed on these solutions, in which their design parameters are optimized in a limited range to calibrate the electrical performances. As can be seen in the third column group, performance loss after physical layout generation is eliminated and the resulting electrical performances satisfy the design objectives. Therefore, it can be concluded that the performance loss is compensated by performing parasitic-aware sizing. Concurrently, layout generation tool achieves an optimum area, and number of fingers for a given template. Figure 6.6 shows the final layout of Solution 1.
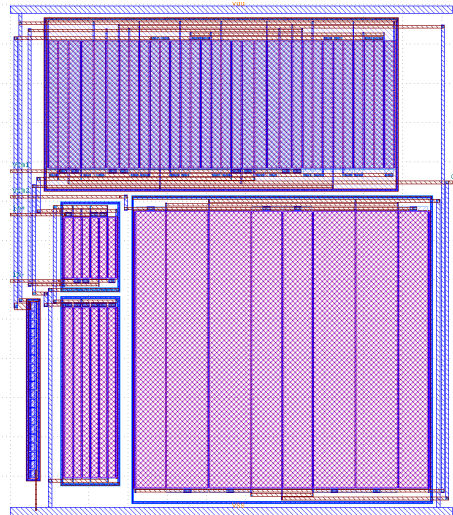

Figure 6.6. Layout corresponding to solution 1.

7. VLSI DESIGN

Physical realization of the circuits are required in order to show the effectiveness of the design automation tools. Therefore, VLSI implementation of a test chip is carried out, which is composed of the circuits obtained at the end of the tool described in Chapter 5.3. Comparator, two stage OTA, and folded cascode amplifier are chosen as design examples. A SPICE input file, including $5mV$ offset and $250MHz$ sampling frequency design constraints is prepared and power is defined as a design objective for comparators. Solution 1, Solution 2, Solution 3, and Solution 4 are obtained corresponding to the $250\mu W$, $750\mu W$, $900\mu W$, and $1500\mu W$ objectives, respectively. In addition, four different solutions are obtained for each Op-Amp topology, where 95% yield, $55dB$ gain, $10kHz$ bandwidth and 60° phase margin are given as design objectives. These four solutions corresponding to the comparator and Op-Amp topologies are placed four times at four different locations to further observe the effects of process variation. Mentor Graphics Layout Editing Software is used for the layout design and DRC, LVS and PEX have been performed by using Mentor utilities.

7.1. Layouts and Post-layout Simulations of Comparators

In general, comparators are considered perfectly symmetric, which means that two sides exhibit identical properties and bias currents. However, in Figure 3.4, pre-amplifier and latch circuits are symmetrical individually, whereas the overall circuit suffers from asymmetry. Even if the circuit is composed of the fully symmetric structure, identical devices suffer from a finite mismatch due to uncertainties in each step of the manufacturing process, which results in an offset in the comparator. As a result, careful layout design is required to reduce the offset as much as possible. The layouts of Solution 1, Solution 2, Solution 3, and Solution 4 are given in Figure 7.1, Figure 7.2, Figure 7.3, and Figure 7.4, respectively. In a simulation setup, $0.7V$ reference voltage and a ramp voltage changing from $0V$ to $1.2V$ are set as inputs of the comparators. The overlapping inputs and positive outputs of the comparators are shown in Figure 7.5, Figure 7.6, Figure 7.7, and Figure 7.8. It can be observed that offsets are obtained

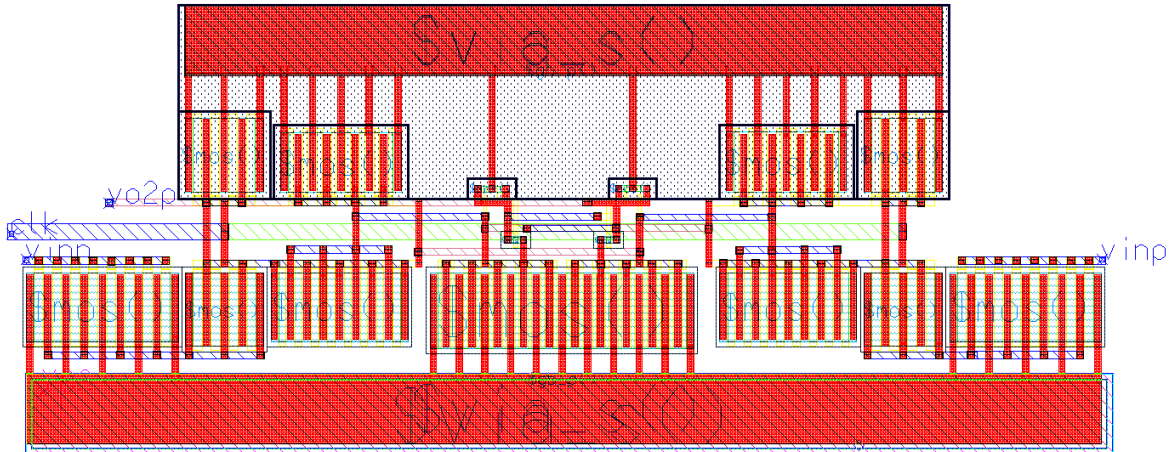as $3.35mV$, $2.39mV$, $2.17mV$, $2.57mV$ for the corresponding solutions.

Figure 7.1. Comparator layout corresponding to solution 1.

7.2. Layouts and Post-layout Simulations of Op-Amps

In this section, a solution that satisfies yield and electrical design objectives was chosen for each topology. The layouts of two stage OTA and folded cascode amplifier are given in Figure 7.9, and Figure 7.10, respectively. $70.07dB$ gain, $10.53kHz$ bandwidth and 69° phase margin were obtained from post-layout simulation, as shown in Figure 7.11. $76.89dB$ gain, $6.93kHz$ bandwidth, 66° phase margin were observed in the post-layout simulation given in Figure 7.12. It is important to emphasize that fingering and rounding of the transistor dimensions were not considered during the optimization process. Therefore, circuit performances are slightly different than that of the SPICE simulations. Finally, overall layouts of the circuits were combined into a test chip, as shown in Figure 7.13
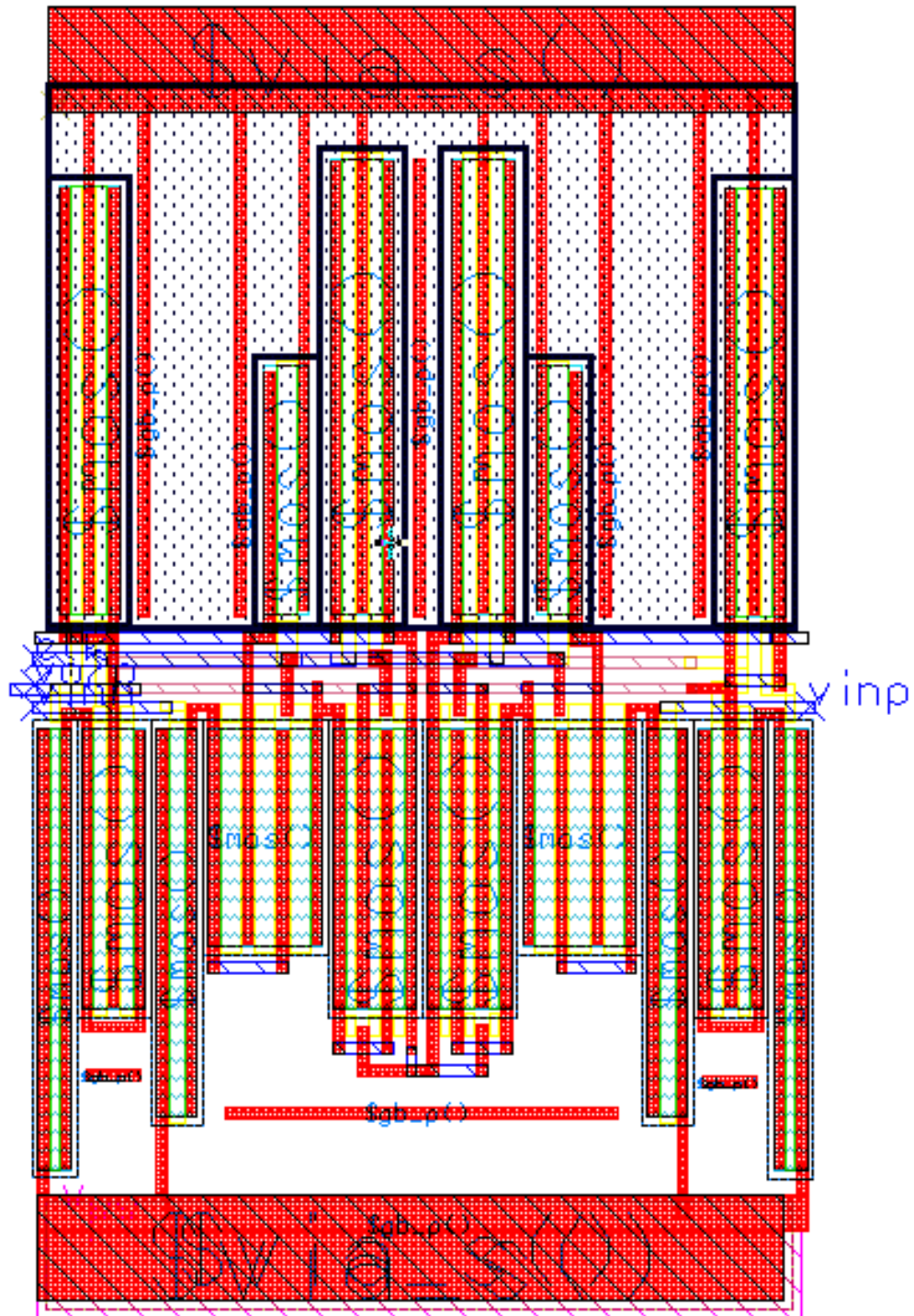
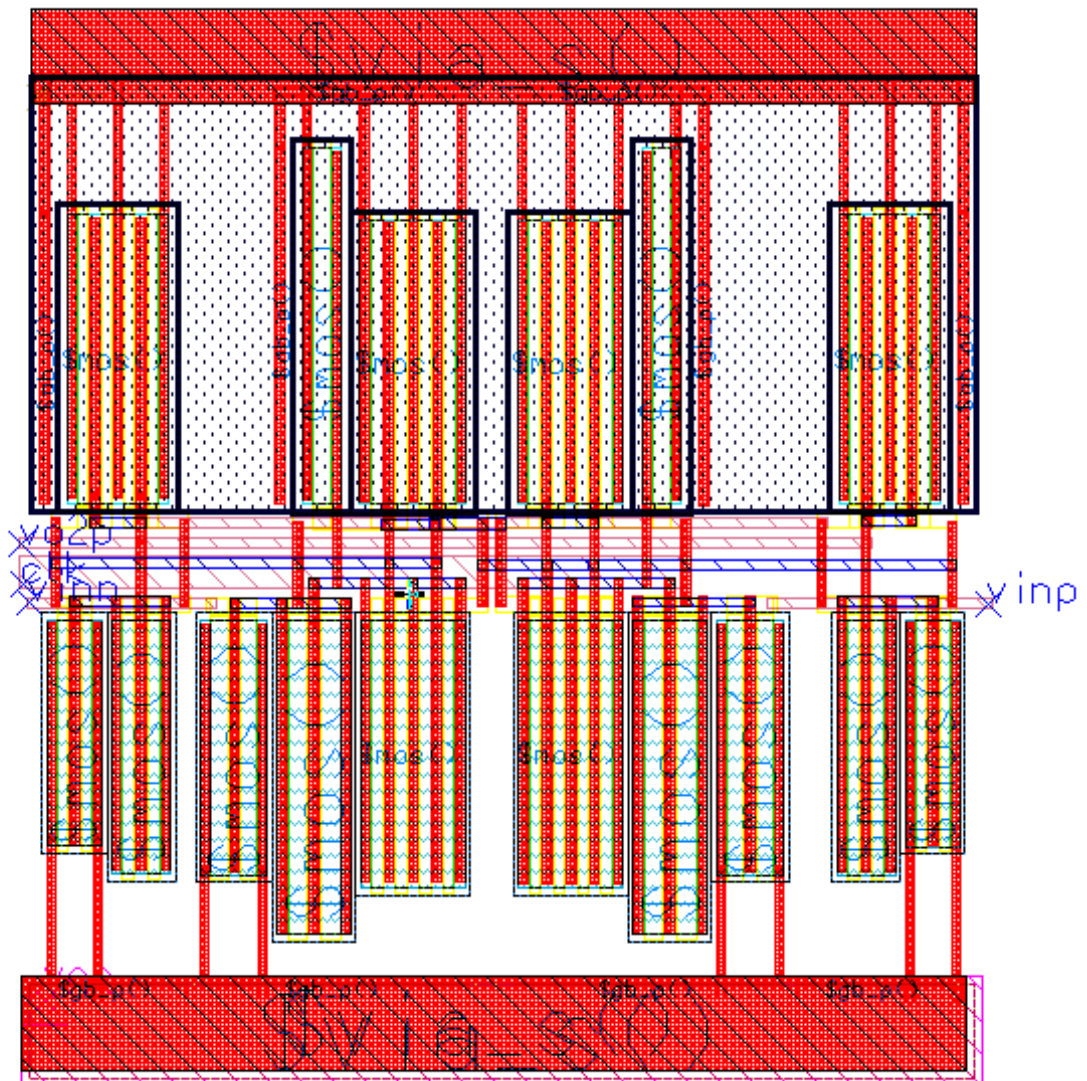Figure 7.2. Comparator layout corresponding to solution 2.

Figure 7.3. Comparator layout corresponding to solution 3.
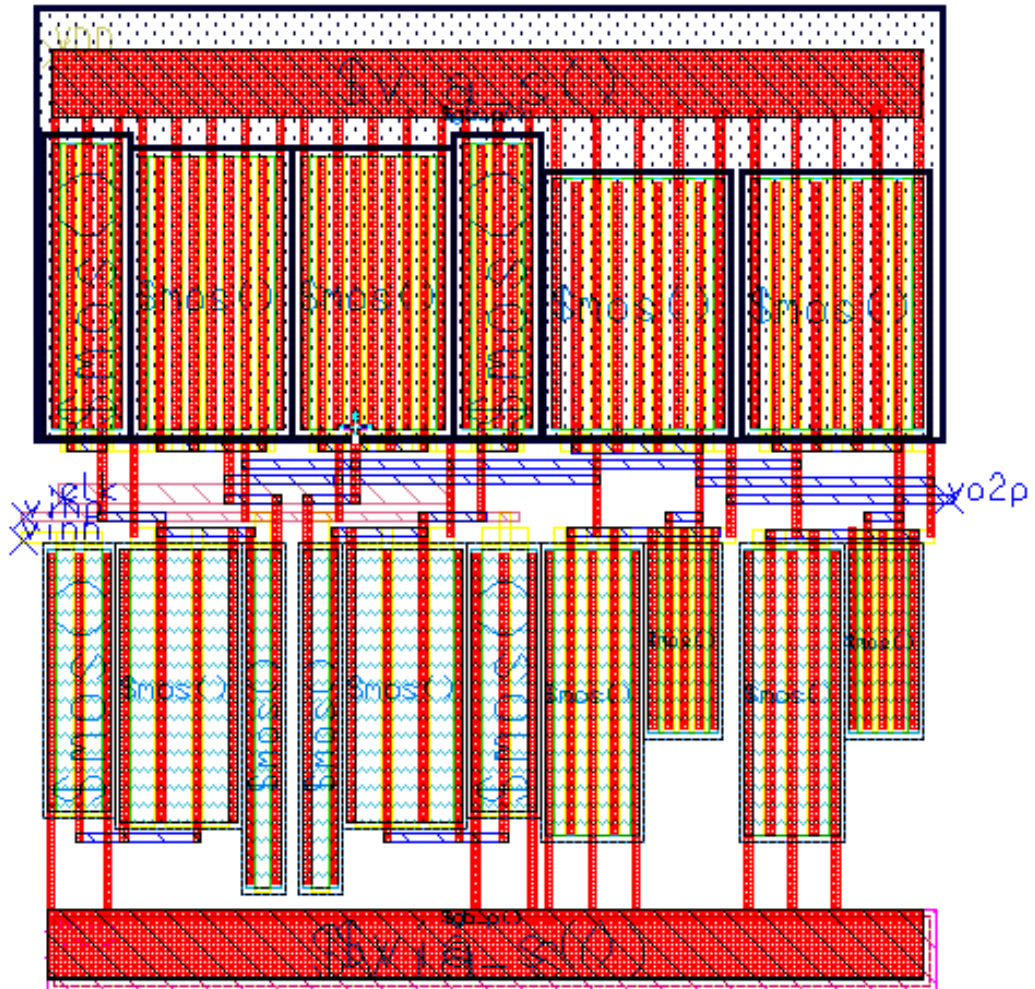
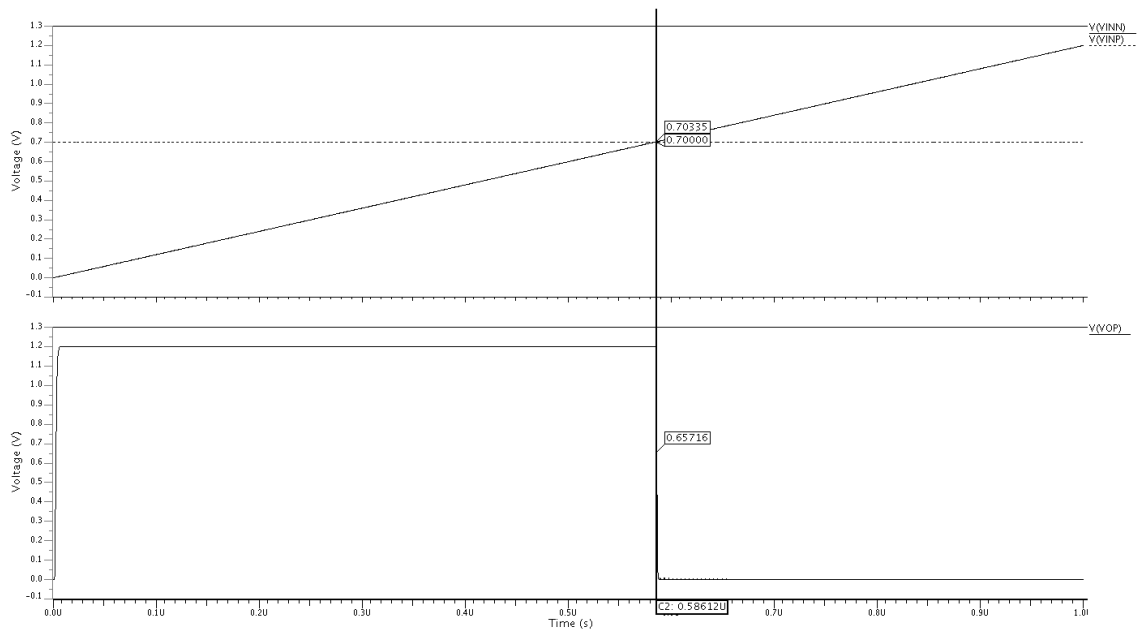Figure 7.4. Comparator layout corresponding to solution 4.

Figure 7.5. Post-layout simulation corresponding to solution 1.

Figure 7.6. Post-layout simulation corresponding to solution 2.

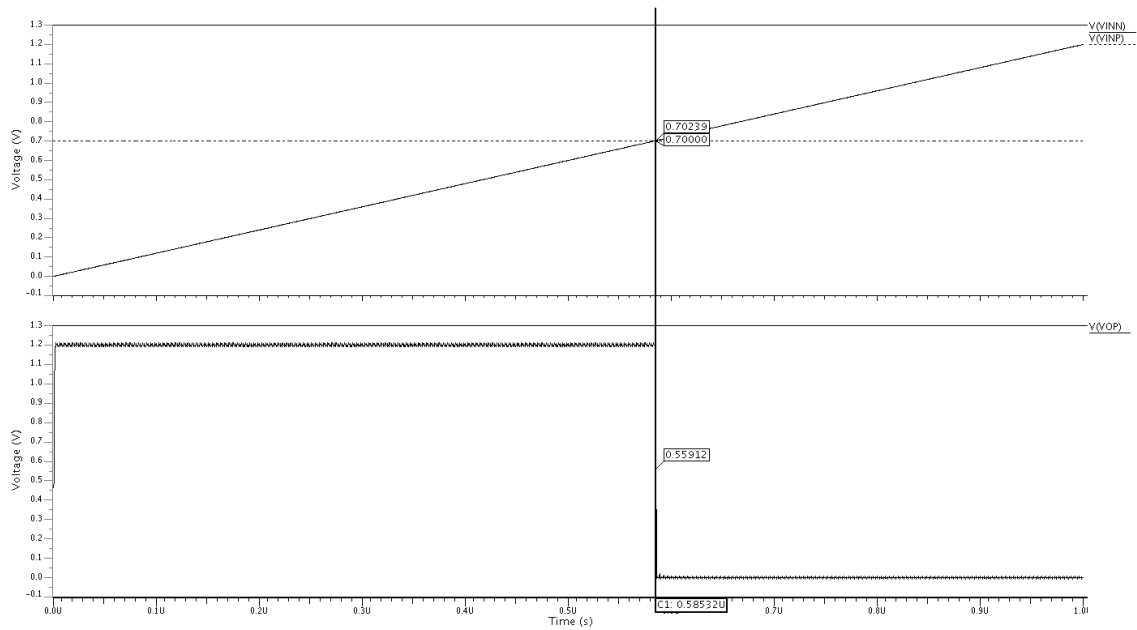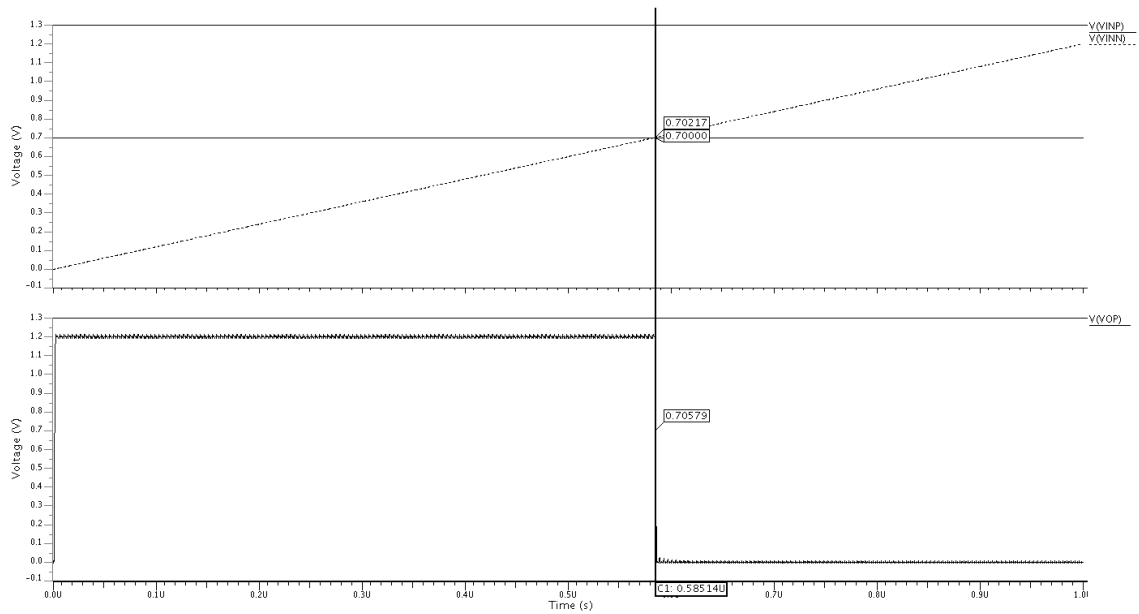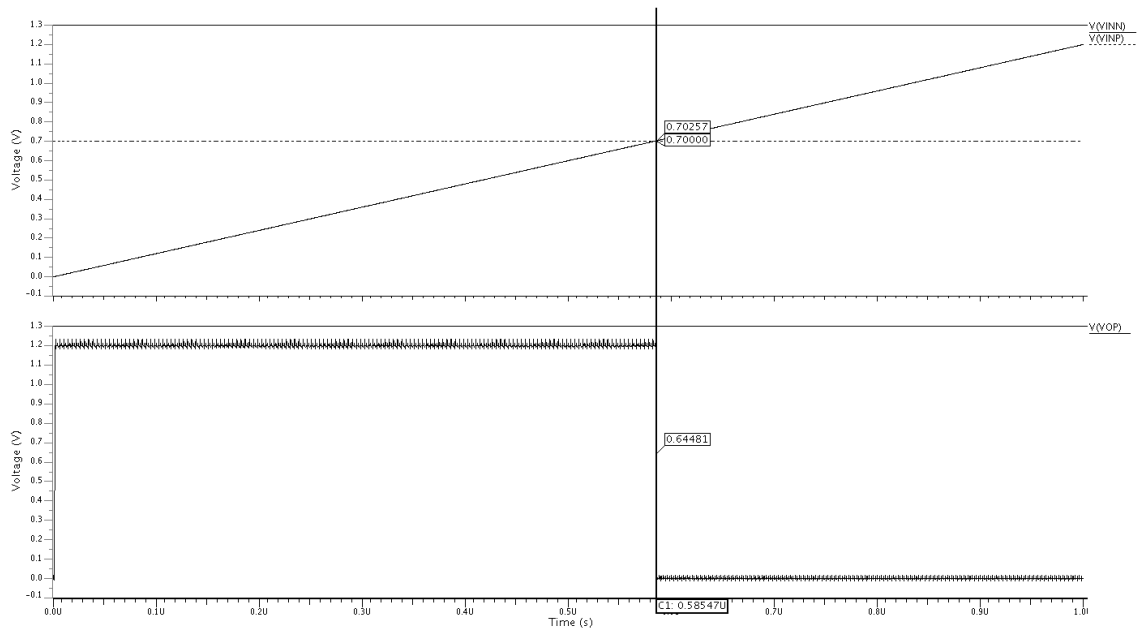Figure 7.7. Post-layout simulation corresponding to solution 3.

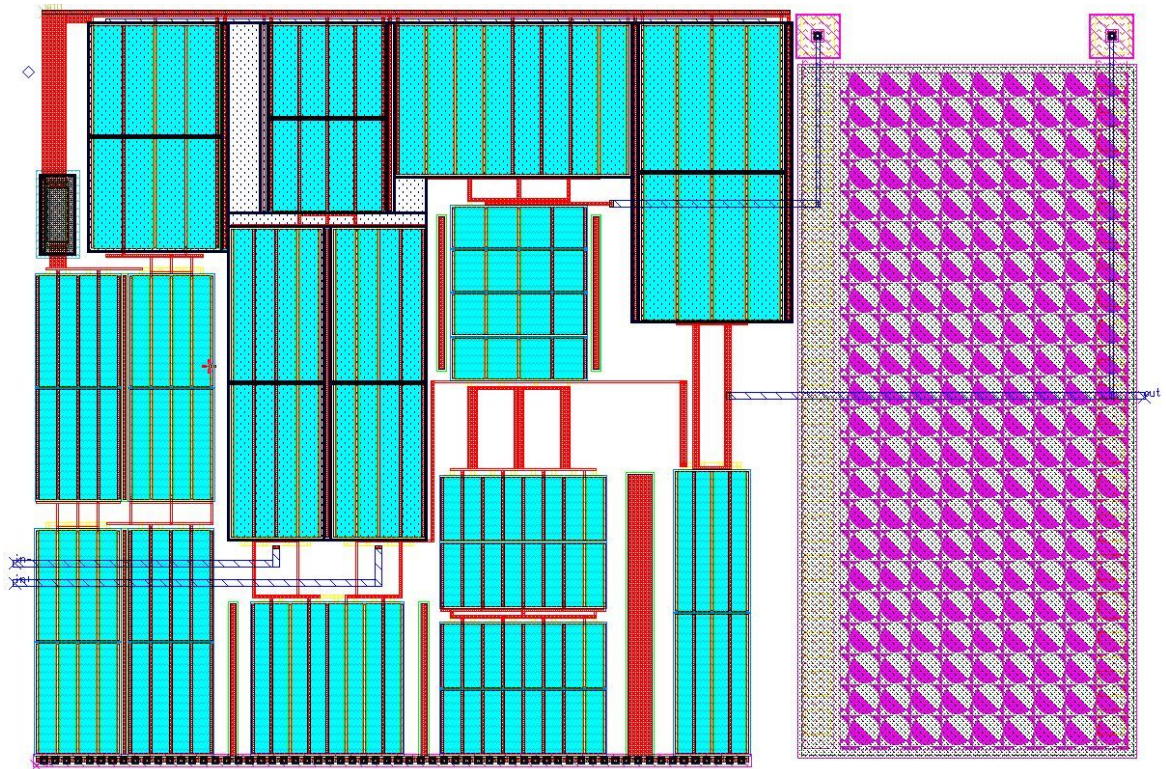Figure 7.8. Post-layout simulation corresponding to solution 4.
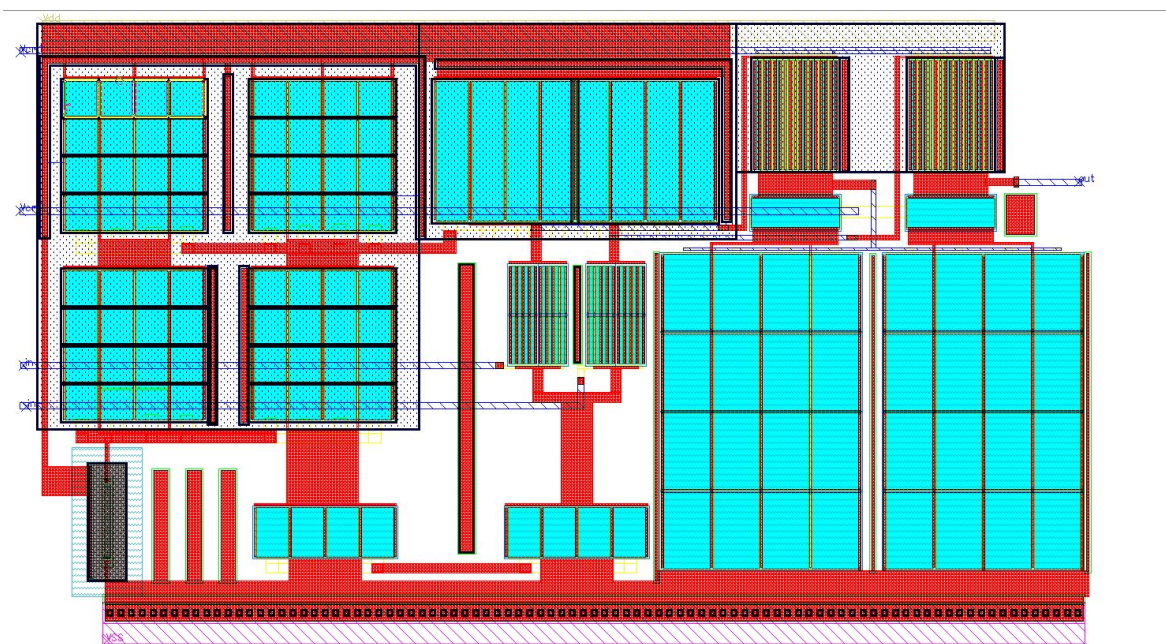
Figure 7.9. Layout of the two stage OTA.

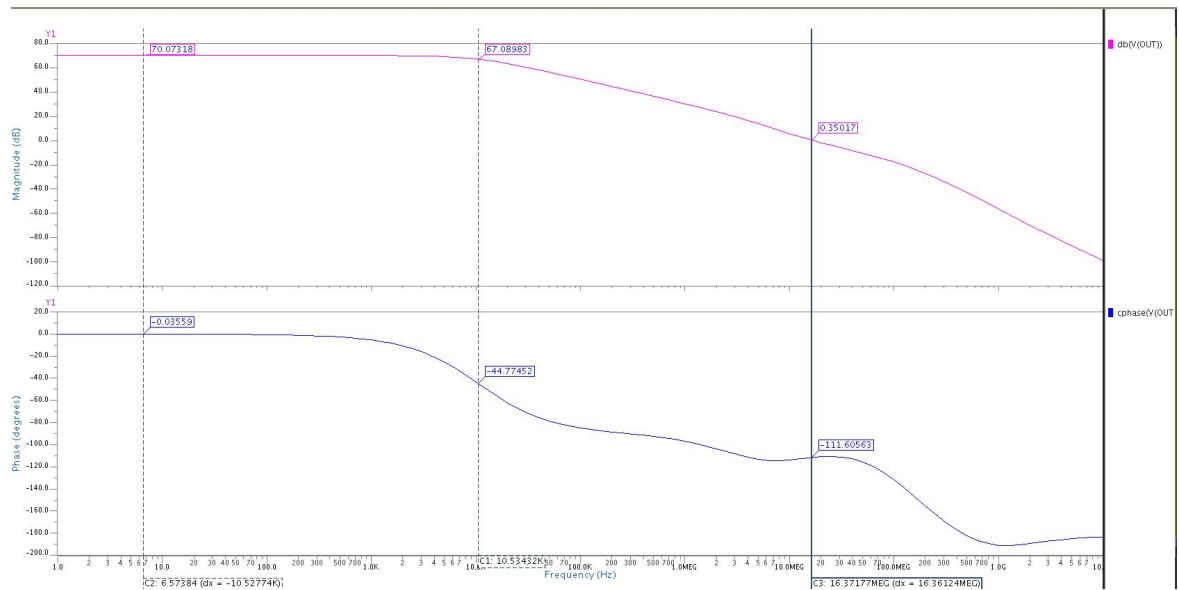Figure 7.10. Layout of the folded cascode amplifier.

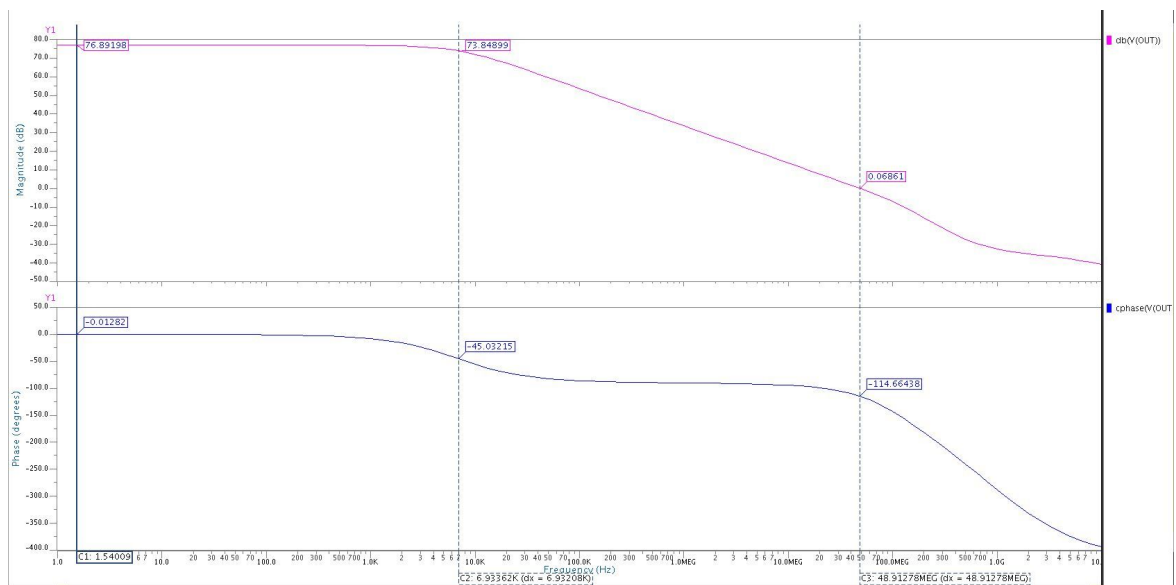Figure 7.11. Post-layout simulation of the folded cascode amplifier.

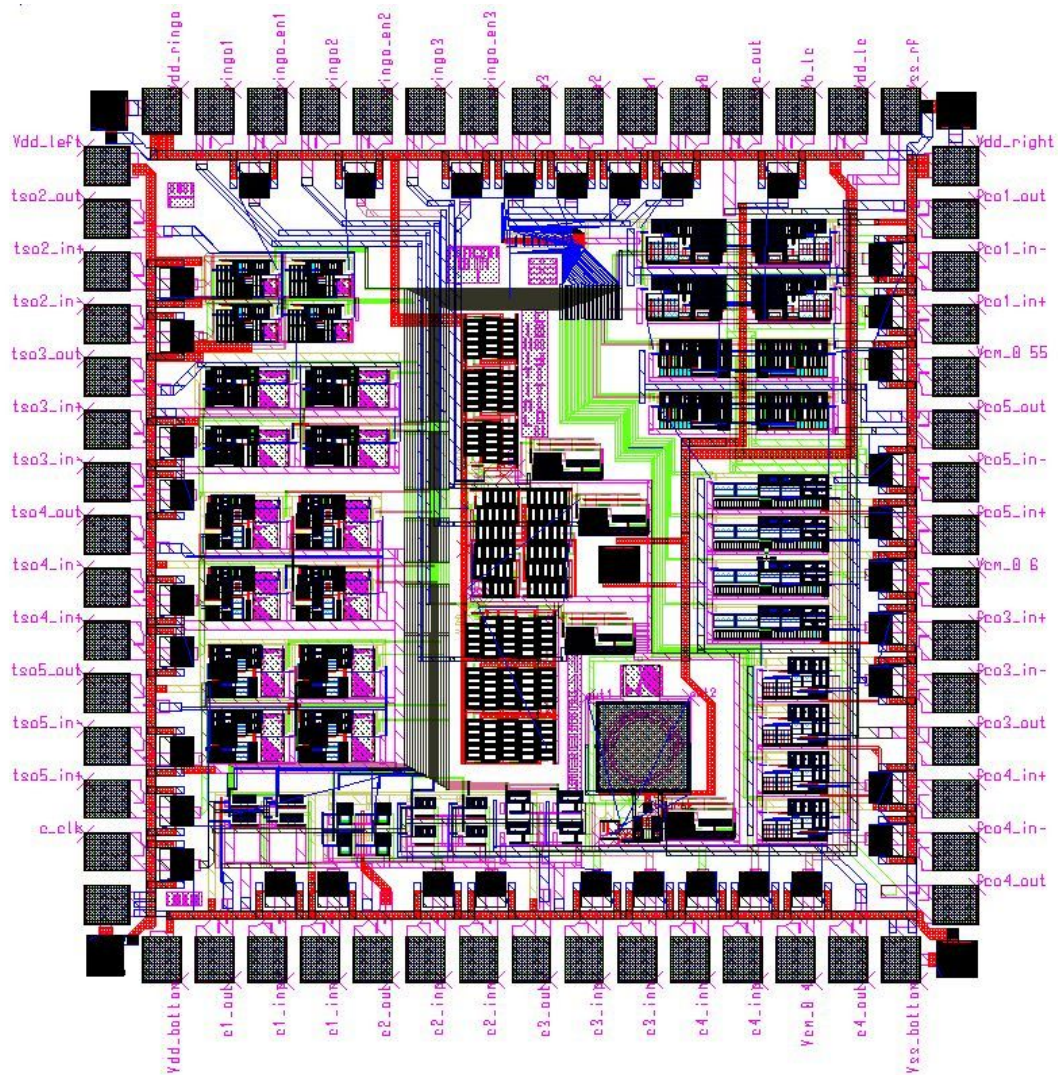
Figure 7.12. Post-layout simulation of the two stage OTA.

Figure 7.13. Layout of the test chip.

8. CONCLUSION

Due to the difference between the scaling of feature size and process tolerances, process variation effects have worsened with developing technology. Worsening process variation effects in sub micron CMOS technology lead to low yields for manufactured ICs. Furthermore, increased secondary effects complicate the analysis of analog circuits; thus, analog circuit design has become a time consuming process. Although several automatic sizing tools have been developed to reduce the design time, yield-aware optimization is still a bottleneck due to the challenging trade-off between the accuracy and efficiency of the yield estimation. In this thesis, a QMC based variability analysis with adaptive sample sizing and automated stopping criterion for yield-aware analog circuit optimization is proposed. Thanks to the deterministic property of the QMC, sampling can be performed iteratively without repeating the previous sample calculations. Moreover, an infeasible solution elimination method is also utilized to increase the efficiency of the optimizer, in which redundant simulations for infeasible solutions are not carried out. The developed tool also includes an optional module, in which a pre-determined simulation budget is shared among the candidate solutions with respect to their yield in order to obtain more accurate yield estimation.

However, a further problem arises when only QMC is used to estimate the yield, in which any statistical error data cannot be given due to the lack of natural variance in QMC. Therefore, scrambled QMC has been introduced, where an artificial variance is created via permuting the order of sample sets randomly. The drawback of the scrambled QMC approach is the requirement of different runs and more simulations for a particular problem that degrades the computational efficiency. To defeat this problem, a hybrid QMC based yield-aware analog circuit optimization tool is proposed. A single QMC is employed for quite accurate yield estimation, which is used to eliminate solutions with low yield values; thus improving the simulation times. In the second part, a couple of scrambled QMC based variability simulations are performed for candidates satisfying the yield constraint and finally yield estimation error can be obtained. Two different OTA topologies were synthesized via the developed tool to

test the proposed methodology and the results were compared with exhaustive MC simulations (10.000 samples). According to the comparison results, the maximum error is found to be 1.46% for two stage OTA and 1.42% for folded cascode amplifier, respectively. Moreover, all MC simulation results (exact yield) are in the confidence interval as promised.

Also, multi-objective optimization tool is developed, since they provide a solution set to the designer and give the chance of selection of a particular design point. However, a further problem arises with worsening variability phenomenon, where some of the optimal design points may shift to the suboptimal region. To overcome this problem, yield-aware optimization tools have been developed, which promise robust Pareto fronts at the end of the optimization. However, a large number of variability simulations are required for the yield estimation when conventional MC analysis is used. To palliate the time efficiency problem without sacrificing the accuracy of the estimation, this thesis proposes a novel multi-objective yield-aware analog circuit optimization tool. The proposed tool utilizes a QMC based yield analysis, in which the estimation error is asymptotically better than that of the conventional MC. Thus, the yield of each individual is accurately estimated during the optimization, and a yield-aware Pareto front can be obtained.

The short time-to-market constraint and excessive design iterations from circuit sizing to layout generation increase the importance of CAD tools. Several methodologies are developed for analog circuit optimization; however, front-end and back-end designs are considered as seperate steps. Therefore, after combining front-end and back-end designs, non-optimal solutions are obtained, which causes degraded circuit performances. In this thesis, a simulation-based circuit sizing tool and a knowledge based layout generation tool are combined to achieve parasitic-aware optimization. As a result, an entire analog optimization tool is proposed to ensure that final solutions are parasitic-aware and are ready for fabrication.

REFERENCES

1. Coello, C. A. C., D. A. Van Veldhuizen and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Vol. 242, Springer, 2002.

2. Gielen, G. G. and R. Rutenbar, "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", *Proceedings of the IEEE*, Vol. 88, No. 12, pp. 1825–1854, 2000.

3. Moore, G. E., "Cramming More Components onto Integrated Circuits", *Proceedings of the IEEE*, Vol. 86, No. 1, pp. 82–85, 1998.

4. Committee, I. R., "International Technology Roadmap for Semiconductors, 2011 Edition", *Semiconductor Industry Association, http://www. itrs. net/Links/2011ITRS/2011Chapters/2011ExecSum.pdf*, 2011.

5. Afacan, E., G. Berkol, F. Baskaya and G. Dundar, "Sensitivity Based Methodologies for Process Variation-Aware Analog IC Optimization", *Microelectronics and Electronics (PRIME), 2014 10th Conference on Ph. D. Research in*, pp. 1–4, IEEE, 2014.

6. Conti, M., P. Crippa, S. Orcioni and C. Turchetti, "Parametric Yield Formulation of MOS IC's Affected by Mismatch Effect", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 18, No. 5, pp. 582–596, 1999.

7. Liu, B., *Computational Intelligence Techniques for Automated Design of Analog and High-Frequency Circuits*, Ph.D. Thesis, Katholieke Universiteit Leuven, 2012.

8. Sönmez, Ö. S. and G. Dündar, "Simulation-Based Analog and RF Circuit Synthesis Using a Modified Evolutionary Strategies Algorithm", *Integration, the VLSI journal*, Vol. 44, No. 2, pp. 144–154, 2011.

9. Zitzler, E., M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", , 2001.

10. Unutulmaz, A., G. Dündar and F. V. Fernández, "LDS-A Description Script for Layout Templates", *Circuit Theory and Design (ECCTD), 2011 20th European Conference on*, pp. 857–860, IEEE, 2011.

11. Harjani, R., L. R. Carley and R. Rutenbar, "OASYS: A Framework for Analog Circuit Synthesis", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 8, No. 12, pp. 1247–1266, 1989.

12. El-Turky, F. and E. E. Perry, "BLADES: An Artificial Intelligence Approach to Analog Circuit Design", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 8, No. 6, pp. 680–692, 1989.

13. Degrauwe, M. G., O. Nys, E. Dijkstra, J. Rijmenants, S. Bitz, B. L. Goffart, E. Vittoz, S. Cserveny, C. Meixenberger, G. Van Der Stappen and H. J. Oguey, "IDAC: An Interactive Design Tool for Analog CMOS Circuits", *Solid-State Circuits, IEEE Journal of*, Vol. 22, No. 6, pp. 1106–1116, 1987.

14. Gielen, G. G., H. C. Walscharts and W. Sansen, "Analog Circuit Design Optimization Based on Symbolic Simulation and Simulated Annealing", *Solid-State Circuits, IEEE Journal of*, Vol. 25, No. 3, pp. 707–713, 1990.

15. Koh, H. Y., C. H. Sequin and P. R. Gray, "OPASYN: A Compiler for CMOS Operational Amplifiers", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 9, No. 2, pp. 113–125, 1990.

16. Harvey, J. P., M. Elmasry and B. Leung, "STAIC: An Interactive Framework for Synthesizing CMOS and BiCMOS Analog Circuits", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 11, No. 11, pp. 1402–1417, 1992.

17. Keding, H., M. Willems, M. Coors and H. Meyr, "FRIDGE: A Fixed-Point Design and Simulation Environment", *Proceedings of the conference on Design, automation and test in Europe*, pp. 429–435, IEEE Computer Society, 1998.

18. Phelps, R., M. Krasnicki, R. Rutenbar, L. R. Carley and J. R. Hellums, "Anaconda: Simulation-Based Synthesis of Analog Circuits via Stochastic Pattern Search", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 19, No. 6, pp. 703–717, 2000.

19. Afacan, E., G. Berkol, A. E. Pusane, G. Dundar and F. Baskaya, "Adaptive Sized Quasi-Monte Carlo Based Yield-Aware Analog Circuit Optimization Tool", *CMOS Variability (VARI), 2014 5th European Workshop on*, pp. 1–6, IEEE, 2014.

20. Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, Vol. 16, John Wiley & Sons, 2001.

21. Darwin, C. and W. F. Bynum, *The Origin of Species by Means of Natural Selection: or, The Preservation of Favored Races in The Struggle for Life*, AL Burt, 2009.

22. Fogel, D. B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, Vol. 1, John Wiley & Sons, 2006.

23. Ishibuchi, H., Y. Nojima and T. Doi, "Comparison Between Single-Objective and Multi-Objective Genetic Algorithms: Performance Comparison and Performance Measures", *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 1143–1150, IEEE, 2006.

24. Yao, L., M. Steyaert and W. M. Sansen, *Low-Power Low-Voltage Sigma-Delta Modulators in Nanometer CMOS*, Vol. 868, Springer Science & Business Media, 2006.

25. Berkol, G., E. Afacan, G. Dundar, A. E. Pusane and F. Baskaya, "A Novel Yield-

Aware Multi-Objective Analog Circuit Optimization Tool", *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, pp. 2652–2655, IEEE, 2015.

26. Zitzler, E., K. Deb and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary computation*, Vol. 8, No. 2, pp. 173–195, 2000.

27. Maaranen, H., K. Miettinen and M. M. Mäkelä, "Quasi-Random Initial Population for Genetic Algorithms", *Computers & Mathematics with Applications*, Vol. 47, No. 12, pp. 1885–1895, 2004.

28. Asenov, A., S. Kaya and A. R. Brown, "Intrinsic Parameter Fluctuations in Decananometer MOSFETs Introduced by Gate Line Edge Roughness", *Electron Devices, IEEE Transactions on*, Vol. 50, No. 5, pp. 1254–1260, 2003.

29. Ye, Y., S. Gummalla, C.-C. Wang, C. Chakrabarti and Y. Cao, "Random Variability Modeling and Its Impact on Scaled CMOS Circuits", *Journal of computational electronics*, Vol. 9, No. 3-4, pp. 108–113, 2010.

30. Afacan, E., G. Berkol, A. E. Pusane, G. Dündar and F. Başkaya, "A Hybrid Quasi Monte Carlo Method for Yield Aware Analog Circuit Sizing Tool", *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pp. 1225–1228, EDA Consortium, 2015.

31. Schwencker, R., F. Schenkel, M. Pronath and H. Graeb, "Analog Circuit Sizing Using Adaptive Worst-Case Parameter Sets", *date*, p. 0581, IEEE, 2002.

32. Okobiah, O., S. P. Mohanty and E. Kougianos, "Ordinary Kriging Metamodel-Assisted Ant Colony Algorithm for Fast Analog Design Optimization", *Quality Electronic Design (ISQED), 2012 13th International Symposium on*, pp. 458–463, IEEE, 2012.

33. Singhee, A., S. Singhal and R. A. Rutenbar, "Practical, Fast Monte Carlo Statisti-

cal Static Timing Analysis: Why and How", *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pp. 190–195, IEEE Press, 2008.

34. Hlawka, E., "Funktionen Von Beschränkter Variatiou in der Theorie der Gleichverteilung", *Annali di Matematica Pura ed Applicata*, Vol. 54, No. 1, pp. 325–333, 1961.

35. Lemieux, C., *Monte Carlo and Quasi-Monte Carlo Sampling*, Springer Science & Business Media, 2009.

36. Pelgrom, M. J., A. C. Duinmaijer and A. P. Welbers, "Matching Properties of MOS Transistors", *IEEE Journal of solid-state circuits*, Vol. 24, No. 5, pp. 1433–1439, 1989.

37. Kullback, S. and R. A. Leibler, "On Information and Sufficiency", *The annals of mathematical statistics*, pp. 79–86, 1951.

38. Chen, C.-H., J. Lin, E. Yücesan and S. E. Chick, "Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization", *Discrete Event Dynamic Systems*, Vol. 10, No. 3, pp. 251–270, 2000.

39. Chen, H.-C., C.-H. Chen and E. Yücesan, "Computing Efforts Allocation for Ordinal Optimization and Discrete Event Simulation", *Automatic Control, IEEE Transactions on*, Vol. 45, No. 5, pp. 960–964, 2000.

40. Owen, A. B., *Randomly Permuted (t, m, s)-Nets and (t, s)-Sequences*, Springer, 1995.

41. Singhee, A. and R. A. Rutenbar, *Novel Algorithms for Fast Statistical Analysis of Scaled Circuits*, Vol. 46, Springer Science & Business Media, 2009.

42. Morohosi, H. and M. Fushimi, "A Practical Approach to the Error Estimation of

Quasi-Monte Carlo Integration", *Monte Carlo and Quasi-Monte Carlo Methods*, Vol. 377, p. 390, 1998.

43. Singhee, A. and R. Rutenbar, "Why Quasi-Monte Carlo is Better Than Monte Carlo or Latin Hypercube Sampling for Statistical Circuit Analysis", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 29, No. 11, pp. 1763–1776, 2010.

44. Berkol, G., A. Unutulmaz, E. Afacan, G. Dundar, F. V. Fernandez, A. E. Pusane and F. Baskaya, "A Two-Step Layout-in-the-loop Design Automation Tool", *New Circuits and Systems (NEWCAS), 2015 IEEE International Symposium on*, IEEE, 2015.

45. Castro-Lopez, R., O. Guerra, E. Roca and F. V. Fernández, "An Integrated Layout-Synthesis Approach for Analog ICs", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 27, No. 7, pp. 1179–1189, 2008.

46. Vancorenland, P., G. Van der Plas, M. Steyaert, G. Gielen and W. Sansen, "A Layout-Aware Synthesis Methodology for RF Circuits", *Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conference on*, pp. 358–362, IEEE, 2001.

47. Dessouky, M. and M. Louerat, "A Layout Approach for Electrical and Physical Design Integration of High-Performance Analog Circuits", *Quality Electronic Design, 2000. ISQED 2000. Proceedings. IEEE 2000 First International Symposium on*, pp. 291–298, IEEE, 2000.

48. Onodera, H., H. Kanbara and K. Tamaru, "Operational-Amplifier Compilation with Performance Optimization", *Solid-State Circuits, IEEE Journal of*, Vol. 25, No. 2, pp. 466–473, 1990.

49. Agarwal, A., H. Sampath, V. Yelamanchili and R. Vemuri, "Fast and Accurate Parasitic Capacitance Models for Layout-Aware", *Proceedings of the 41st annual*

Design Automation Conference, pp. 145–150, ACM, 2004.

50. Ranjan, M., W. Verhaegen, A. Agarwal, H. Sampath, R. Vemuri and G. Gielen, "Fast, Layout-Inclusive Analog Circuit Synthesis Using Pre-Compiled Parasitic-Aware Symbolic Performance Models", *Proceedings of the conference on Design, automation and test in Europe-Volume 1*, p. 10604, IEEE Computer Society, 2004.

51. Liao, Y.-C., Y.-L. Chen, X.-T. Cai, C.-N. Liu and T.-C. Chen, "LASER: Layout-Aware Analog Synthesis Environment on Laker", *Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI*, pp. 107–112, ACM, 2013.

52. Zhang, L. and Z. Liu, "Directly Performance-Constrained Template-Based Layout Retargeting and Optimization for Analog Integrated Circuits", *Integration, the VLSI journal*, Vol. 44, No. 1, pp. 1–11, 2011.

53. Chen, Y.-L., Y.-C. Ding, Y.-C. Liao, H.-J. Chang and C.-N. Liu, "A Layout-Aware Automatic Sizing Approach for Retargeting Analog Integrated Circuits", *VLSI Design, Automation, and Test (VLSI-DAT), 2013 International Symposium on*, pp. 1–4, IEEE, 2013.

54. Youssef, S., F. Javid, D. Dupuis, R. Iskander and M.-M. Louerat, "A Python-Based Layout-Aware Analog Design Methodology for Nanometric Technologies", *Design and Test Workshop (IDT), 2011 IEEE 6th International*, pp. 62–67, IEEE, 2011.

55. Lourenço, N., A. Canelas, R. Póvoa, R. Martins and N. Horta, "Floorplan-Aware Analog IC Sizing and Optimization Based on Topological Constraints", *Integration, the VLSI Journal*, Vol. 48, pp. 183–197, 2015.

56. Habal, H. and H. Graeb, "Constraint-Based Layout-Driven Sizing of Analog Circuits", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 30, No. 8, pp. 1089–1102, 2011.

57. Unutulmaz, A., G. Dündar and F. V. Fernández, "Template Coding with LDS and Applications of LDS in EDA", *Analog Integrated Circuits and Signal Processing*, Vol. 78, No. 1, pp. 137–151, 2014.

58. Unutulmaz, A., G. Dündar and F. V. Fernández, "Area Optimization on Fixed Analog Floorplans using Convex Area Functions", *Proceedings of the conference on design, automation and test in europe*, pp. 1843–1848, EDA Consortium, 2013.