Synthesis of Manufacturable Analog Circuits

Tamal Mukherjee, L.R. Carley and R.A Rutenbar

Electrical and Computer Engineering Dept
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract[†]

We describe a synthesis system that takes operating range constraints and inter- and intra- circuit parametric manufacturing variations into account while designing a sized and biased analog circuit. Previous approaches to CAD for analog circuit synthesis have concentrated on nominal analog circuit design, and subsequent optimization of these circuits for statistical fluctuations and operating point ranges. Our approach simultaneously synthesizes and optimizes for operating and manufacturing variations by mapping the circuit design problem into an Infinite Programming problem and solving it using an annealing within annealing formulation. We present circuits designed by this integrated synthesis system, and show that they indeed meet their operating range and parametric manufacturing constraints.

1 Introduction

Improvements in performance and level of integration have led to the replacement of printed circuit boards by a single IC, in many cases, and to an increase in the presence of analog functionality in traditionally digital application specific integrated circuits (ASICs). A wide range of analog synthesis strategies has emerged to design these circuits. These strategies range from solving both the topology selection and device sizing/biasing problems simultaneously to solving them in tandem; from using circuit simulators for evaluating circuit performance, to behavioral equations predicting circuit performance; from searching the design space with optimization, to using a set of inverted behavioral equations with a restricted search space. The problem with these approaches is that most of them synthesize circuits considering only a nominal operating point and a nominal process point. At best, existing approaches allow the expert synthesis tool creator to pre-select specific operating and process points for performance evaluation.

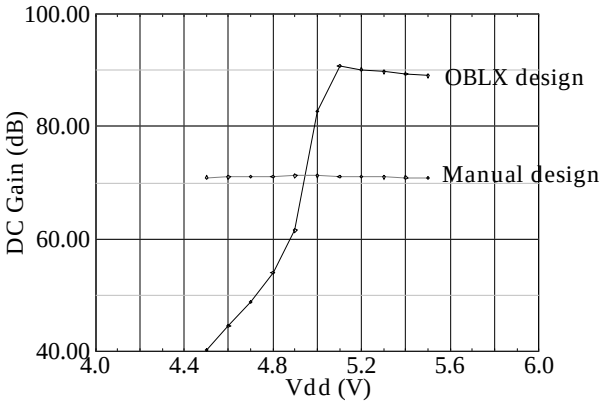
Because integrated circuits are sensitive to parametric fluctuations in the manufacturing process, design with a nominal set of manufacturing process parameters is insufficient. In addition, all circuits are sensitive to fluctuations in their operating conditions (*e.g.*, power supply voltages and temperature). Traditionally, designers first generate a nominal design using topologies known to be relatively tolerant of operating range and parametric manufacturing variations, then improve the design's manufacturability using worst-case methods. These methods evaluate the design at other sets of values for the process and operating point. The circuit is redesigned (*i.e.*, sizing and bias point is modified) to increase the number of sets at which all the performance specifications are met.

The advent of computer-based circuit simulation led to the first studies of computer-based circuit design, *e.g.*, [31]. Since then we have seen approaches to optimization-based nominal circuit design [24], and even analog circuit synthesis [4][11][12][13][18][21][26]. We have also seen approaches to statistical IC design [6][8][15][33] and even some recent approaches that include operating ranges [1][5]. The goal of analog circuit synthesis tools is to eliminate many of the circuit designer's most tedious tasks. The goal of yield optimization tools is to improve an already well-designed circuit prior to fabrication. While both sets of tools were aimed at helping the designer, they both solved just half the problem. The analog circuit synthesis tools often create designs that are at the edge of the performance space, whereas a good human designer using a familiar topology knows exactly how to over-design to ensure adequate yield. The yield optimization tools significantly improve the yields of good manual designs but the automatically synthesized circuit is often a bad starting point for the gradient-based post-synthesis yield optimization tools. As will be demonstrated in the results section, device sizing is frequently driven by manufacturing or operating variations. Synthesis based only on the nominal operating and manufacturing point can result in dramatically different device sizes/biasing. This may be incompatible with the assumptions of most yield optimization tools.

In Figure 1 we show an example of the dc gain performance of a manual operational amplifier and one synthesized by [25] (marked OBLX). The synthesis was done at the nominal power supply level of $V_{dd} = 5.0V$, and the circuit definitely meets the design specification of

†. This work was supported by the Semiconductor Research Corporation.

Figure 1 Variation of dc gain of Manual and Synthesized Design with V_{dd}



$DCgain = 70dB$, at that nominal operating point. However, several devices are biased near the edge of saturation and a small decrease in power supply voltage dramatically decreases the gain. The circuit is highly sensitive to that operating point variable, and therefore provides poor initial yields and yield gradients needed for the post-design yield optimization phase.

We present a unified design framework in which inter-chip and intra-chip statistical parametric fluctuations in the fabrication line, operating point fluctuations, and analog circuit synthesis strategies are combined to form a system that can synthesize realistically manufacturable analog circuits. This is the first integration of intra-chip fluctuations and operating point fluctuations in the same algorithm of which we are aware. It is also the first integration of practical manufacturability concerns into an analog circuit synthesis tool of which we are aware. In this paper we first discuss earlier approaches to analog circuit synthesis, then introduce the concept of using an Infinite Programming formulation for the analog circuit synthesis problem. We review some approaches to solving the Infinite Programming problem, and present an annealing-based approach. We then present three circuits designed using the annealing approach to Infinite Programming, and finally offer some concluding remarks.

2 Review of Analog Circuit Synthesis

The system presented in this paper is based on ASTRX/OBLX [26]. In this section we will briefly review the main characteristics of the ASTRX/OBLX synthesis strategy. The primary assumption of this strategy is that the computer's time is less valuable than the designer's time. So the goal is to force the computer rather than the designer to do the bulk of the synthesis work. ASTRX/OBLX therefore can determine the ac performance of the circuit without requiring designer-supplied equations, thus it can perform "equation-free" synthesis of linear circuits [25]. A constraint language [26] is used by ASTRX/OBLX to support arbitrary large-signal and transient specifications such as output swing and slew rate. Since the acac equations in the earlier synthesis strategies tended to be both more cumbersome and more numerous than the non-linear performance equations, this trade-off of "equation-free" linear design and "equation-based" non-linear design has been successful at designing the widest range of analog circuits to date [26][27][28].

The key ideas in the ASTRX/OBLX synthesis strategy are described below:

Synthesis via Optimization: the synthesis problem is mapped onto a constrained optimization formulation that is solved in an unconstrained fashion. As in [18][21][24], the circuit design problem is mapped to the non-linear constrained optimization problem (NLP) of equation (1), where $\underline{u}$ is the vector of independent variables—geometries of semiconductor devices or values of passive circuit components—we wish to change to determine circuit performance, and $\underline{x}$ is the vector of state variables; $f(\underline{u}, \underline{x})$ is a set of objective functions that codify performance specifications the designer wishes to optimize, *e.g.*, area or power; and $\underline{h}(\underline{u}, \underline{x}) = 0$ and $\underline{g}(\underline{u}, \underline{x}) \le 0$ are each a set of constraint functions that codify specifications that must be beyond a specific goal, *e.g.*, $60dB - Gain(\underline{u}, \underline{x}) \le 0$. Scalar weights, w_i, balance competing objectives. The decision variables can be described as a set $\underline{u} \in U_P$, where U_P is the set of allowable values for $\underline{u}$.

$$\left.\begin{aligned} min_{\underline{u}} \quad & z = \sum_{i=1}^{k} w_i \cdot f_i(\underline{u}, \underline{x}) \\ \text{s.t.} \quad & \underline{h}(\underline{u}, \underline{x}) = 0 \\ & \underline{g}(\underline{u}, \underline{x}) \le 0 \\ & \underline{u} \in U_P \end{aligned}\right\} (NLP) \qquad (1)$$

To allow the use of simulated annealing, in ASTRX/OBLX this constrained optimization problem is converted to an unconstrained optimization problem with the use of additional scalar weights.

$$C(\underline{u}) = \sum_{i=1}^{k} w_i f_i(\underline{u}, \underline{x}) + \sum_{i=1}^{l} w_i g_i(\underline{u}, \underline{x}) + \sum_{i=1}^{m} w_i h(\underline{u}, \underline{x}) \qquad (2)$$

As a result, the goal becomes minimization of a scalar cost function, $C(\underline{u})$, defined by equation (2). The key to this formulation is that the minimum of $C(\underline{u})$ corresponds to the circuit design that best matches the given specifications. Thus, the synthesis task is divided into two sub-tasks: evaluating $C(\underline{u})$ and searching for its minimum.

Asymptotic Waveform Evaluation: AWE [30] is augmented with some simple, automatically generated analytical analyses to convert AWE transfer functions into circuit performances. AWE is a robust, efficient approach to anal-

ysis of arbitrary linear RLC circuits that for many applications is several orders of magnitude faster than SPICE. By matching the initial boundary conditions and the first $2q-1$ moments of the actual RLC circuit transient response to a reduced q-pole model, AWE can predict ac circuit response using a reduced complexity model valid for frequencies below a specifiable limit. AWE is a general-purpose simulation technique that can be applied to any linear(ized) circuit and yields accurate results without manual analysis.

Simulated annealing [17] is the optimization engine which will drive the search for a circuit solution; it provides robustness and the potential for global optimization in the face of many local minima. Because annealing incorporates controlled *hill-climbing* it can escape local minima and is essentially starting-point independent. Annealing also has other appealing properties including: the inherent robustness of the algorithm in the face of discontinuous cost functions, and the ability to optimize without derivatives. Further, with the increasing research effort on annealing control mechanisms, it is now achieving competitive run-times on problems for which tuned heuristic methods exist [19].

Encapsulated device evaluators, comprise a compiled database of industrial models for active devices that provides the accuracy of a general-purpose simulator while making the synthesis tool completely independent of low-level device modelling concerns [20]. Unfortunately, there is no longer any alternative to using industrial-strength device models in a practical circuit synthesis system. High-performance circuits rely on the device performance characteristics achievable in aggressive technologies. Simplified models of the current-voltage relationships at the terminals of these devices fail to capture many of these important characteristics. The ASTRX/OBLX synthesis strategy has a library of encapsulated device evaluators which hides all aspects of the device's representation and performance and provides a query interface for standard device large and small-signal information. In this manner, the models are completely independent of the synthesis system and can be easily replaced or altered.

A relaxed dc-formulation [20] which avoids a CPU-intensive dc operating point solution after each perturbation of the circuit design variables. Since encapsulated models must be treated numerically, as in circuit simulation, an iterative algorithm such as Newton Raphson is required. For synthesis, such an approach has two important pitfalls: First, it is a well-known source of convergence problems; and second, it consumes a substantial amount of CPU time that would be wasted on intermediate circuit designs that are later discarded. Instead, the Kirchhoff laws that are implicitly solved during dc biasing in a circuit simulator are explicitly formulated and included in the constraint functions in the optimization problem. Just as optimization goals are formulated, such as meeting gain or bandwidth constraints, now dc-correctness is formulated as yet another goal that needs to be met. This relaxed approach allows the optimizer to visit many more circuits within a given time, modelling each a little less accurately.

In ASTRX/OBLX, these ideas are combined to create an architecture that provides a fully automated path from an un-sized analog circuit topology and a set of performance specifications to a completed, synthesized circuit. This path is comprised of two phases:

Compilation by ASTRX: Compilation generates a performance prediction module that maps $\underline{u}$, the component and dc bias values in the circuit, to the performance metrics specified by the user. In effect, ASTRX generates code that implements a cost function, $C(\underline{u})$, carefully constructed so that its minimum value occurs when $\underline{u}$ specifies a circuit design that best meets the input specifications. To evaluate this cost function, ASTRX will compile in the appropriate links to the encapsulated device evaluators and AWE. Because of the relaxed dc-formulation, ASTRX must also derive the dc correctness constraints that will enforce Kirchhoff's laws for the input topology and encode them in the cost function.

Solution by OBLX: This cost function code is then compiled and linked to OBLX, the solution library, which uses simulated annealing to numerically solve for the minimum of $C(\underline{u})$, thereby designing the circuit. The variables in $\underline{u}$ are mapped directly to aspects of the evolving circuit design, such as device sizes and node voltages. However, because node voltage values must clearly be continuous to determine an accurate bias point while device sizes can reasonably be regarded as discrete quantities, OBLX manipulates a mix of continuous and discrete variables. Because of this mix, the annealer's *move-set*, the set of allowable perturbations on the present state, is quite complex. In addition the conventional annealing moves where the discrete variables are *atomically* perturbed, we employ *gradient-based* moves in which the entire vector of continuous node voltages is perturbed (using the Newton Raphson algorithm [26]). To dynamically determine the correct mix of move sizes throughout the annealing process, an adaptation of a method from Hustin [16] originally developed for discrete annealing problems is used. OBLX also employs other sophisticated control mechanisms to provide reliable solutions over varying circuits and specifications: A dynamic weighting system automatically adjusts the weights on the terms in $C(\underline{u})$, freeing the user from this task and a general purpose cooling schedule derived from Lam [19] and Swartz [34] controls the overall annealing process.

3 Infinite Programming approach to Analog Circuit Synthesis

In this section we will expand the non-linear constrained optimization formulation in ASTRX/OBLX to a

non-linear infinite programming formulation that considers operating range and parametric manufacturing variations. Integrated circuit designers often guess worst-case range limits from experience for use during the initial design phase. Once they have generated a sized schematic for the nominal operating point (*i.e.*, bias, temperature) and process parameters, they test their design across a wide range of process and operating points (*worst-case set*). Typically the initial design meets specifications at several worst-case sets, but needs to be modified to ensure that the specifications are met at the rest of the worst-case sets.

Unfortunately, in an optimization-based approach to circuit design, the optimization will usually find any holes left open by a poor or incomplete set of constraints. Therefore the initial design produced by the optimization may be a bad starting point for the remainder of the design (a complete statistical IC design [6]). Current statistical IC design algorithms assume that the initial circuit meets all the specifications at the nominal process point. As operating range specifications are part of the design problem, circuit synthesis tools need to generate designs that meet the specifications not only for the nominal set of process parameters, but also over the entire operating point range.

Most designs have a $\pm 10\%$ range specification on the power supply voltage, leading to an operating range $V_{dd} \in [4.5, 5.5]\,V$ in a 5V process. As we have seen in Figure 1, the circuit designer considered the sensitivity of these circuit performance metrics to the operating range variable V_{dd}. In contrast, the automatic design was blind to V_{dd} except at one point, $V_{dd} = 5.0V$. Similar graphs for the other performances in many analog circuits (from simulation, as well as from data books) show that:

- Low V_{dd} designs are the ones most likely to fail to meet the performance specifications, so there is a need to consider the additional constraint of $V_{dd} = 4.5V$ in the mathematical program used to design the circuit.
- Not all of the performance parameters are monotonic functions of the operating point. Therefore a mechanism to find the worst-case point in the range for each performance function is needed. However, since many are monotonic, the infinite set of constraints can be replaced by a single constraint at a worst-case corner point, although a mechanism for doing this automatically is needed.

To investigate the effect of introducing operating ranges to the NLP model equation (1), let us consider the example of dc gain, a circuit performance metric, and the power supply voltage range, an operating range: the dc gain needs to be larger than 60dB for every power supply voltage value in the range $4.5 \le V_{dd} \le 5.5$. This can be written as in equation (1), where $\underline{u}$ is the vector of designable parameters, and ξ is a new variable (in this example only a scalar), to represent the operating range:

$$60dB - Gain(\underline{u}, \underline{x}, \xi) \le 0 \quad \forall \xi \in [4.5, 5.5]\,V.$$

Since every single voltage in the given range needs to be investigated, this single operating range constraint adds an *infinite* number of constraints to the mathematical program. Heittich, Fiacco and Kortanek present several papers in [9][14] which discuss non-linear optimization problems where some constraints need to hold for a given range of an operating range variable. These problems, and the one just presented, are called *Semi-Infinite Programs* due to the finite number of objectives and an infinite number of constraints. When there is an infinite number of objective functions (due to the presence of a range variable in the objective function), the mathematical program is called an *Infinite Program*. The complete mathematical program can now be re-written as the *Non-Linear Infinite Program* (NLIP) shown in equation (3).

$$\begin{aligned} & min_{\underline{u}} \quad z = \sum_{i}^{k} w_i \cdot f(\underline{u}, \underline{x}, \xi) \\ \text{s.t.} \quad & \left.\begin{array}{ll} \underline{h}(\underline{u}, \underline{x}, \underline{\xi}) = 0 & \forall \underline{\xi} \in \Xi \\ \underline{g}(\underline{u}, \underline{x}, \underline{\xi}) \le 0 & \forall \underline{\xi} \in \Xi \\ \underline{u} \in U_P & \end{array}\right\} (NLIP) \end{aligned} \quad (3)$$

where Ξ is the vector set of operating point ranges and statistical fluctuations. If we let Ξ include the region in the disturbance space where the probability of the disturbance is, say, less than 3σ from nominal, then the design problem NLIP implies that the design has to meet all the specifications for any disturbance that is less than 3σ. Further details of the formulations can be found in [23], and similar formulations have been presented in [1][5]. This infinite programming formulation looks at yield as a constraint, and device mismatch disturbance [6][29] can also be incorporated into this framework, as will be shown by our examples.

4 A Conceptual Infinite Programming Algorithm

In this section, we will review the solution of a simple non-linear infinite program. Since it is easier to discuss a semi-infinite program, and it illustrates all the critical points, we will actually consider a simple semi-infinite program (an infinite program can be considered to be a mini-max semi-infinite program, and mini-max problems are mainstream optimization problems [10]). Linear and non-linear infinite programs have received significant attention in the optimization literature [9][14]. Consider the problem *P*, which is a simplified version of *NLIP* in equation (3) (without the state variables and equality constraints, and just a single objective that is independent of the worst-case variables and a single constraint):

$$P: min_{\underline{u} \in \underline{U}_P} \{ \quad f(\underline{u}) \mid g(\underline{u}, \underline{\xi}) \le 0, \forall \underline{\xi} \in \Xi \quad \} \quad (4)$$

Now, let $W(\underline{u})$ denote the problem:

$$W(\underline{u}): \quad max_{\underline{\xi} \in \Xi} g(\underline{u}, \underline{\xi}) \tag{5}$$

A simple conceptual algorithm begins with a $\underline{\xi}_0 \in \Xi$ and solves

$$P_0: \quad min_{\underline{u} \in \underline{U}_p} \{f(\underline{u}) \mid g(\underline{u}, \underline{\xi}_k) \le 0, k = 0\} \tag{6}$$

to obtain $\underline{u}_0$ (note that P_0 has only one constraint, and so can be solved using standard NLP algorithms). Then, $\underline{\xi}_1$ is computed by solving $W(\underline{u})$; if $g(\underline{u}_0, \underline{\xi}_k) \le 0$, it stops; otherwise it proceeds to solve

$$P_1: \quad min_{\underline{u} \in \underline{U}_p} \{f(\underline{u}) \mid g(\underline{u}, \underline{\xi}_k) \le 0, k = 0, 1\} \tag{7}$$

for a $\underline{u}_1$, etc. We can generalize this sequence of "outer problems", labelled as P_i, in which there are only $i + 1$ constraints; they can be written as:

$$P_i: \quad min_{\underline{u} \in \underline{U}_p} \{f(\underline{u}) \mid g(\underline{u}, \underline{\xi}_k) \le 0, k = 0, 1, \ldots, i\} \tag{8}$$

Consider the feasible set of the problems P_i:

$$F_i = \{\underline{u} \mid g(\underline{u}, \underline{\xi}_k) \le 0, k = 0, 1, \ldots, i\} \tag{9}$$

and Problem P:

$$F = \{\underline{u} \mid g(\underline{u}, \underline{\xi}_k) \le 0, \forall \underline{\xi} \in \Xi\} \tag{10}$$

The sequence of feasible sets satisfy

$$F_0 \supset F_1 \supset \ldots \supset F \tag{11}$$

thus the sequence of objective function values have the following property (since more and more constraints are added as k increases):

$$f(\underline{u}_0) \le f(\underline{u}_1) \le \ldots \le f(\underline{u}_i) \le \ldots \le f(\underline{u}^*) \tag{12}$$

where $\underline{u}^*$ is any solution to Problem P. Eaves and Zangwill, in [7], develop a theory to show that any accumulation point $\hat{\underline{u}}$ of $\{\underline{u}_i\}_1^\infty = 0$ is in F, and solves Problem P. They also show the constraint $g(\underline{u}, \underline{\xi}_k) \le 0$ can be dropped from Problem P_i if:

- $g(\underline{u}_{i-1}, \underline{\xi}_k) \le 0$ and $f(\underline{u}_{i-1})$ is sufficiently larger than $f(\underline{u}_k)$, and
- the next solution $\underline{u}_i$ satisfies $f(\underline{u}_i) \ge f(\underline{u}_{i-1})$.

This constraint-dropping scheme suggests that the growth of complexity of Problem P_i can be slowed, but in the absence of convexity, the solution of P_i is only a local minima, so the sequence $\{f(\underline{u}_i)\}$ is not necessarily monotonically increasing, so the Eaves and Zangwill constraint-dropping scheme cannot be applied. In the domain of circuit synthesis, complicated combinations of nonlinear device equations make it extremely difficult, if not impossible, to make any statements about convexity. Hence we use a simulated annealing optimization method, and exploit its hill-climbing abilities in the solution of both P_i and $W(\underline{u})$. Annealing has been shown to be empirically robust on many problems with complex non-convex cost surfaces [32].

5 Infinite Programming in ASTRX/OBLX

In this section we extend the NLP formulation in ASTSX/OBLX to a non-linear infinite programming formulation. First let us consider a direct implementation of the conceptual algorithm into ASTRX/OBLX. OBLX currently solves the problem P_0, and can easily be extended to solve the outer optimization problems P_i. Each OBLX run takes from a few minutes to a few hours, making it prohibitive to consider this alternative.

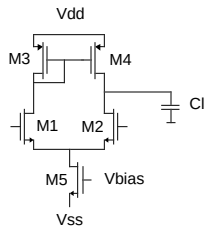
Instead, we solve for the worst-case constraints in the middle of the annealing run, specifically, at each point when the annealing temperature related to the outer problem is reduced. This leads to a single annealing run to solve all the outer optimization problems P_i (albeit slightly longer). Inside this single annealing run, at every change in the temperature, the number of worst-case sets increases depending on the worst-case inner optimization. This approach is the middle ground between solving the inner optimization problems ($W(\underline{u})$) at each perturbation of the outer annealing problem (P_i), and solving the inner optimization between each annealing runs as suggested by the direct implementation of the conceptual scheme presented in the previous section.

Figure 1 shows that it is not always necessary to do an inner optimization, since the function $g(\underline{u}_i, \underline{\xi})$ is often a one-dimensionally monotonic function of $\underline{\xi}$. Thus the first part of the solution of $W(\underline{u})$ should involve a test for monotonicity. We use a large-scale sensitivity computation to determine monotonicity, and pick the appropriate corner of the worst-case range from there. This test can be applied to operating point variables which have box constraints on them ($\Xi_{iL} \le \xi_i \le \Xi_{iU}$, where Ξ_{iL} is the lower bound and Ξ_{iU} is the upper bound for the dimension in which $g(\underline{u}_i, \underline{\xi})$ is one dimensionally monotonic). Applying such bounds to the statistical variables will lead to conservative designs. It will be left up to the user to trade-off between applying these bounds and getting conservative designs quickly, or actually doing an inner optimization over the space of statistical design variables to get a less conservative design.

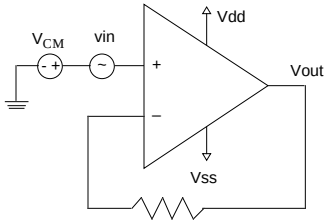
6 Results

We applied the above annealing approach to solving the non-linear infinite programming formulation of the analog circuit synthesis problem to a small operational transconductance amplifier (OTA) cell, and to a large folded cascode amplifier cell. We compare these results with the original ASTRX/OBLX [26] designs to show that it is indeed important to take parametric manufacturing variations and operating point variations into account during analog circuit synthesis. In both circuits, we first added the V_{dd} op-

Figure 2 An Operational Transconductance Amplifier



Simple OTA Circuit



Test jig for Simple OTA

erating range as the only worst-case variable to consider. In addition, we also considered global variations in transistor geometries, global parametric variations in threshold voltages, and intra-chip parametric variations in the threshold voltages of matched devices on the OTA to show that the NLIP formulation can incorporate both inter- and intra-chip statistical fluctuations in the form of yield as a constraint.

Figure 2 shows the Simple OTA circuit and the test-jig used to simulate the circuit in HSPICE [22] for the results presented below. There are 6 design variables: the width and length of the differential-pair transistors (M1 and M2), the width and length of the current mirror transistors (M3 and M4), the width and length of the tail current source transistor (M5), and the Vbias voltage. For the NLIP formulation, we added the V_{dd} operating point variable as a worst-case variable. We compare the designs generated by looking at the performance graphs across the $4.5 \le V_{dd} \le 5.5$ operating range. Figure 3 shows the dc gain performance of the ASTRX/OBLX and NLIP formu-

Figure 3 OTA dc gain performance

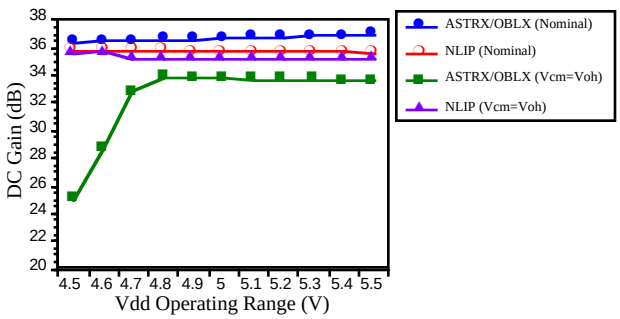
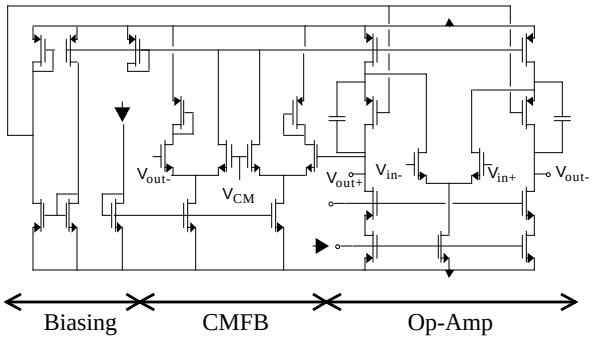
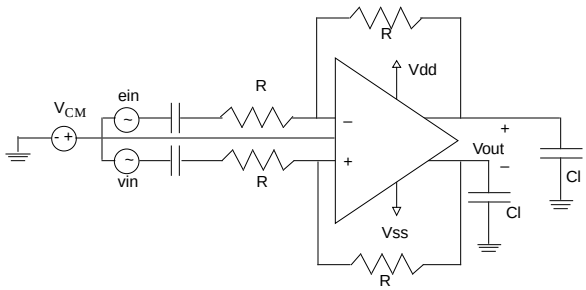


Figure 4 The Folded Cascode Amplifier



Folded Cascode Circuit



Test Jig for Folded Cascode

lation design, simulated at $V_{CM} = 2.5V$ (labeled Nominal) and $V_{CM} = V_{OH}$, the specified output high voltage (3.75V). Note that at the nominal operating point ($V_{dd} = 5.0V$, $V_{CM} = 2.5V$) the dc gain of the ASTRX/OBLX design is no more sensitive to the operating point than is the nominal NLIP design. This illustrates that even adding small-change sensitivity constraints at the nominal operating point would not improve the design. The actual worst-case gain of this circuit will occur when the common mode of the input voltage (called V_{CM} here) is at its highest specified value, in this case the highest output voltage (V_{OH}) since the test-jig is configured for unity-gain feedback and V_{dd} is at its lowest value. It is clear from the graph that it is necessary to use the NLIP formulation to ensure that the dc gain is relatively insensitive to the operating range. Since the worst-case point is an operating range corner, the designer can actually ask ASTRX/OBLX to design for that corner by pre-specifying $V_{dd} = 4.5V$, and $V_{CM} = V_{OH}$ instead of their nominal values. However, it is not always possible *apriori* to identify the worst-case corner in a larger example (with more worst-case variables), and in some cases, the worst-case point can occur within the operating range (*e.g.*, variation in output voltage with temperature for a temperature-compensated band-gap reference).

The same experiment was repeated with the folded cascode amplifier shown in Figure 4. In this design there are twelve designable widths and lengths, a designable com-

pensation capacitance and two dc bias voltages. Again the single operating point variable V_{dd} was used. We simulated the ASTRX/OBLX and the NLIP designs' output swing across the V_{dd} operating range (shown in Figure 5). The output swing of the amplifier is a strong function of the region of operation of the output transistors (M3, M4, M5 and M6). The output swing is obtained by using a large-signal ac input, and determining the output voltage at which the output transistors move out of saturation (which will cause clipping in the output waveform). Compared to the OTA, this is a much more difficult design, hence the output swing specification of 2.0V is just met by both the ASTRX/OBLX design (at the nominal power supply voltage of 5.0V) and the NLIP design (across the entire operating range).

The ASTRX/OBLX design fails to meet the output swing specification for the lower half of the operating range ($V_{dd} < 5.0V$). This is a common problem of synthesis tools. For an optimal design, it biased the circuit so that the output transistors were at the edge of saturation, and a slight decrease in the V_{dd} voltage resulted in their moving out of saturation, hence the output swing falls below the 2.0V specification.

In our final experiment we again reconsider the OTA circuit of Figure 2. This time we introduce global variations in transistor geometries (*e.g.*, ΔL of the p and n devices); global parametric variations in threshold voltages (V_{FB}, the flat-band voltage of the p and n devices); and intra-chip parametric variations in the threshold voltages of matched devices (ΔV_{FB} of the p and n devices, using a simple mismatch model proposed by [29]). Note, the intra-chip parametric variations are particularly challenging because their amplitude depends on the design variables — these variations are roughly proportional to $1/(\sqrt{WL})$.

In this run there were 6 circuit design variables, and 6 worst-case variables. We expect to see that the device sizes will be larger to minimize the effect of the mismatch in geometry and threshold voltage. It should be obvious that larger geometries reduce the sensitivity to the ΔL variation. In addition, larger devices are less sensitive to the ΔV_{FB} mismatch than are smaller devices [29], again pushing for larger designs.

Figure 5 Folded Cascode Output Swing across V_{dd}

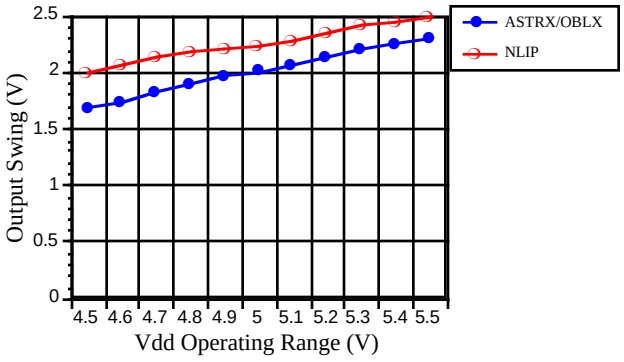


Figure 6 OTA Random Offset Voltage

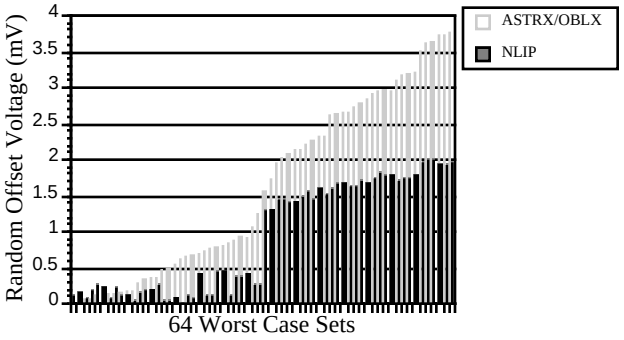


Figure 6 shows the random component of the input referred offset voltage (the systematic component of both designs is less than 0.5mV), at the 64 worst-case corners of the 6 variables used in the design. The ASTRX/OBLX design can have a random offset voltage of up to 4mV. The non-linear infinite programming formulation of the analog circuit synthesis problem was set up with a random offset voltage specification of 2mV. In the graph we have sorted the corners by the size of the random offset voltage in the ASTRX/OBLX design. From the graph it is clear that about half of the 64 corners need to be considered by the formulation. It is also clear that the optimization has reduced the random offset voltage only as much as was needed to meet the specification. For the half of the worst-case corners that are not active constraints, the NLIP optimization returns a design whose random offsets are actually greater than that of the ASTRX/OBLX design for that corner. While these corners are currently inactive, at a different sizing and biasing for the circuit, those corners can easily become active. This prevents the designer using ASTRX/OBLX from *apriori* defining the list of worst-case points. In an incompletely specified problem, the optimization will find a solution that might violate a constraint that was not specified.

Table 1 Device sizes (in microns) of the OTA and parametric manufacturing variations

Design	ASTRX/ OBLX	NLIP
W_{M1}	26	71
L_{M1}	2.7	3.3
W_{M3}	4.8	15
L_{M3}	4.6	8.6
W_{M5}	6.6	58
L_{M5}	2.1	15
CPU Time on RS6000/550	10 min	900 min

Our approach can dynamically determine the actual worst-case points, using the inner annealing and the large-scale sensitivity of the cost function terms to the worst-case variables, hence it limits the designer's responsibility to just providing the worst-case ranges. Note that the execution time of this run for this prototype implementation is only 90 times more than the nominal design done by ASTRX/OBLX. This is driven because we are evaluating 64 times as many circuits (so each move is longer), and because the annealer needs to do more moves to efficiently search the search space.

7 Conclusions

In this paper we have integrated analog circuit synthesis with worst-case analysis of both parametric manufacturing and operating point variations. This integration has been used to design circuits that are manufacturable. By showing that an automated system can generate circuits that can meet some of the critical concerns of designers (operating ranges and inter- and intra-chip parametric variations), we believe that we have taken a significant step towards the routine use of analog synthesis tools in an industrial environment.

Acknowledgments

The authors would like to acknowledge Dr. Emil Ochotta for his help in describing the working details of ASTRX and OBLX. The authors would also like to acknowledge Prof. Steve Director for his direction in the early stages of this work.

References

[1] K.J. Antriech, H.E. Graeb, and C.U. Wieser, "Circuit Analysis and Optimization Driven by Worst-Case Distances", *IEEE Trans. CAD*, vol. 13, no. 1, pp. 57-71, Jan. 1994.

[2] G.L. Bilbro and W.E. Snyder, "Optimization of Functions with Many Minima", *IEEE Trans on Systems, Man and Cybernetics*, vol. 21, no. 4, pp. 840-849, July/August 1991.

[3] I.D. Coope and G.A. Watson, "A Projected Lagrangian Algorithm for Semi-Infinite Programming", *Mathematical Programming*, pp. 337-356, 1985.

[4] M.G.R. Degrauwe *et al*., "Towards an analog system design environment," *IEEE JSSC*, vol. SC-24, no. 3, June 1989.

[5] A. Dharchoudhury and S.M. Kang, "Performance-Constrained Worst-Case Variability Minimization of VLSI Circuits", *Proc. 30th ACM/IEEE DAC*, pp. 154-158, 1993.

[6] S.W. Director, P. Feldmann and K. Krishna, "Optimization of Parametric Yield: A Tutorial", *Proc IEEE CICC*, pp. 3.1/1-8, May 1992.

[7] B.C. Eaves and W.I. Zangwill, "Generalized Cutting Plane Algorithms", *SIAM Journal of Control*, vol. 9, pp 529-542, 1971.

[8] P. Feldmann and S.W. Director, "Integrated Circuit Quality Optimization using Surface Integrals", *IEEE Trans. CAD*, vol. 12, no. 12, pp. 1868-1879, Dec. 1993.

[9] A.V. Fiacco and K.O. Kortanek, eds., *Semi-Infinite Programming and Applications, Lecture notes in Economics and Mathematical Systems 215*, Springer-Verlag, 1983.

[10] R. Fletcher, *Practical Methods of Optimization*, Wiley, 1990.

[11] G. Gielen, *et al*., "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE JSSC*, vol. SC-25, no. 3, June 1990.

[12] R. Harjani, R.A. Rutenbar and L.R. Carley, "OASYS: a framework for analog circuit synthesis," *IEEE Trans. CAD*, vol. 8, no. 12, Dec. 1989.

[13] J. P. Harvey, *et al*., "STAIC: An Interactive Framework for Synthesizing CMOS and BiCMOS Analog Circuits," *IEEE Trans. CAD*, Nov. 1992.

[14] R. Heittich, ed., *Semi-Infinite Programming, Lecture notes in Control and Information Sciences 15*, Springer-Verlag, 1979.

[15] D.E. Hocevar, P.F. Cox and P. Yang, "Parametric yield optimization for MOS circuit blocks", *IEEE Trans. CAD*, vol. 7, no. 6, June 1988.

[16] S. Hustin and A. Sangiovanni-Vincentelli, "TIM, a new standard cell placement program based on the simulated annealing algorithm", presented at IEEE Physical Design Workshop on Placement and Floorplanning, Hilton Head, SC, April 1987.

[17] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, 13 May 1983.

[18] H.Y. Koh, C.H. Sequin, and P.R. Gray, "OPASYN: a compiler for MOS operational amplifiers," *IEEE Trans. CAD*, vol. 9, no. 2, Feb. 1990.

[19] J. Lam and J.M. Delosme, "Performance of a New Annealing Schedule," *Proc. 25th ACM/IEEE DAC*, pp. 306-311, 1988.

[20] P.C. Maulik and L.R. Carley, "Automating Analog Circuit Design using Constrained Optimization Techniques", *Proc. IEEE ICCAD*, pp. 390-393, Nov. 1991.

[21] P.C. Maulik, L.R. Carley and D.J. Allstot, "Sizing of Cell-Level Analog Circuits Using Constrained Optimization Techniques", *IEEE JSSC*, vol. SC-28, no. 3, March 1993.

[22] Metasoft Corp. HSPICE manual, 1990.

[23] T. Mukherjee, "Incorporating Manufacturing and Operating Point Related Considerations into Optimization-based methods for Analog Circuit Synthesis", *Phd Prospectus*, Carnegie Mellon University, Dec. 1992.

[24] W. Nye, *et al*., "DELIGHT.SPICE: an optimization-based system for the design of integrated circuits," *IEEE Trans. CAD*, vol. 7, no. 4, April 1988.

[25] E.S. Ochotta, R.A. Rutenbar, and L.R. Carley, "Equation-Free Synthesis of High-Performance Linear Analog Circuits," *Proc. 1992 Brown/MIT Adv. Res. VLSI and Parallel Systems Conf.*, pp 129-143, The MIT Press, 1992.

[26] E.S. Ochotta, R.A. Rutenbar, and L.R. Carley, "ASTRX/OBLX: Tools for Rapid Synthesis of High Performance Analog Circuits," *Proc. 31st ACM/IEEE DAC*, pp. 24-30, June 1994.

[27] E.S. Ochotta, L.R. Carley and R.A. Rutenbar, "Analog Circuit Synthesis for Large, Realistic Cells: Designing a Pipelined A/D Converter with ASTRX/OBLX", *Proc. IEEE CICC*, pp. 365-368, May 1994.

[28] E.S. Ochotta, "Synthesis of High-Performance Analog Cells in ASTRX/OBLX," Ph.D thesis, Carnegie Mellon University, 1994.

[29] M.J.M. Pelgrom, A.C.J. Duinmaijer, and A.P.G. Welbers, "Matching properties of MOS transistors", *IEEE JSSC*, vol. SC-24, no.5, Oct. 1989.

[30] R.A. Rohrer, *et al*., "AWE Inspired," *Proc. IEEE CICC*, pp. 18.1.1-8, May 1993.

[31] R.A. Rohrer, "Fully Automated Network Design by Digital Computer: Preliminary Considerations", *Proceedings of the IEEE*, vol. 55, no. 11, pp. 1929-1939, Nov. 1967.

[32] F. Romeo and A. Sangiovanni-Vincintelli, "A Theoretical Framework for Simulated Annealing", *Algorithmica*, 6: 302-345, 1991.

[33] M.A. Styblinski and L.J. Opalski, "Algorithms and software tools of IC yield optimization based on fundamental fabrication parameters", *IEEE Trans. CAD*, vol. 5, no. 1, Jan. 1986.

[34] W. Swartz and C. Sechen, "New Algorithms for the Placement and Routing of Macrocells," *Proc. IEEE ICCAD*, pp. 336-339, Nov. 1990.