Stochastic Optimization Approach to Transistor Sizing for CMOS VLSI Circuits

Sharad Mehrotra Paul Franzon[1] Wentai Liu[2]

Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

Abstract

A stochastic global optimization approach is presented for transistor sizing in CMOS VLSI circuits. This is a direct search strategy for the best design among feasible ones, with the designer determining when the search is stopped. Through examples, we show the power of this technique in quickly obtaining very good designs, for skew minimization problems.

1 Introduction

In designing high performance CMOS circuits, it is necessary to properly size the various transistors in the circuit, in order to meet performance requirements. There are two basic approaches for transistor sizing that have been explored by various researchers. The first approach, applied primarily to delay optimization of combinational circuits, involves developing a simplified model of signal delay through a CMOS gate, either analytically [2], or by macromodels based on simulations [4]. Then, the delay of the entire circuit is computed using this model. The transistor sizing problem is formulated as an optimization problem with the objective of minizing the delay, as predicted by the model. Some optimization problems have a special form to guarantee efficiency and accuracy of the optimization process, e.g. the posynomial objective function, used in [2]. Otherwise, a nonlinear optimization method is required [4]. The second approach involves coupling a circuit simulator to a nonlinear optimization tool (or tool set). For example, Ochotta et. al. [7] employ an augmented asymptotic waveform evaluation technique to evaluate the behavior of each circuit visited by a simulated annealing program. In Delight.Spice [6], a set of nonlinear optimization programs are integrated with the SPICE circuit simulator.

The first, 'equation-based', approach, though effective for delay optimization, is difficult to generalize to other problems, such as skew optimization. Accounting for input data vector and process variation with an analytical model is very difficult. The only solution is to evaluate each sizing scheme through circuit simulation, for each data vector and process corner. This makes objective computation a very expensive task, even for very small sized circuits. The second, 'simulation-based' approach has the combined computational burden of running a nonlinear optimization program and multiple circuit simulations each time the objective function is evaluated. There are some additional hindrances in employing a conventional nonlinear optimization program for this task:

1. Gradient information is very difficult to obtain. Though there are numerical optimization techniques which do not require explicit gradient information, these techniques tend to be slow.
2. Transistor sizes can only be varied in certain quanta. Most numerical optimization techniques operate on a continuous parameter range. Hence the final solution might not be designable.
3. The optimization task is not interactive, and it is difficult for the engineer to use his or her judgement in guiding the optimizer.
4. The optimization routines look for strict local minima. Usually, the designer is interested only in obtaining an approximation to a globally optimal solution. For this, the optimizer has to be run from multiple, random, initial solutions.

In view of these difficulties, we present a new approach to transistor sizing in small, high performance circuits. This approach is based on stochastic modeling of the circuit responses of interest. It is a direct search for the best design among feasible ones. The designer has direct control over the number of simulations conducted, and the search process can be stopped any time the designer is satisfied with the best solution produced thus far. The stochastic model helps in identifying the most promising design based on the existing information about the problem. Only the most promising designs are simulated.

[1] supported by NSF under MIP-901704 and a Young Investigator award and by ARPA under contract DAAH04-94-G-003.

[2] supported by NSF under MIP-9212346.

Hence simulations are organized naturally and efficiently.

2 Optimization by stochastic modeling

Consider the following unconstrained optimization problem:

$$\min_x \quad \zeta(x) \quad x \in A \subset R^d \tag{1}$$

where x is a d dimensional vector, A is a finite subset of R^d. $\zeta(x)$ is the objective function whose value at any $x \in A$ can be determined only through an expensive simulation. Besides this, there is very little information about the objective function. Suppose the objective function is perceived to be continuous and 'smooth', but not unimodal. In such a situation, it is reasonable to approximate $\zeta(x)$ by a simpler function and to perform an optimization on this simplified function. However, a global polynomial approximation is inappropriate; information is lost in fitting the model to data, unless the degree of the polynomial is as large as the data set. A better approach to function approximation is needed.

Recently, stochastic models have been proposed to capture complex objective functions [9]. With this approach, the value of the unknown function at each point in A is assumed to be a random variable. Then, the unknown function itself is a *sample path of a stochastic function.* In the general case, a stochastic function $\phi(x)$ is defined by a family of multidimensional probability distributions $F_{x_1,\ldots,x_m}(y_1,\ldots,y_m) = P(\phi(x_i) < y_i, i = 1,\ldots,m)$. For example, if this distribution is joint Gaussian, then the stochastic function is described by the *a priori* average function $\mu(x)$ and covariance $\sigma(x_i, x_j)$. If k values of the stochastic function are known, e.g. $\phi(x_i) = \zeta(x_i), i = 1,\ldots,k$ then the *conditional distribution* of $\phi(x)$ at any x is normal with the mean value

$$\begin{aligned} m_k(x \mid \phi(x_i) \quad &= \quad \zeta(x_i)) = \mu(x) + \\ &\quad (\sigma(x,x_1),\ldots,\sigma(x,x_k))R_k^{-1} \\ &\quad (\zeta(x_1) - \mu(x_1),\ldots,\zeta(x_k) - \mu(x_k))^T, \end{aligned} \tag{2}$$

and variance,

$$\begin{aligned} s_k^2(x \mid \phi(x_i) \quad &= \quad \zeta(x_i)) = \sigma(x,x) - \\ &\quad (\sigma(x,x_1),\ldots,\sigma(x,x_k)) \\ &\quad R_k^{-1}(\sigma(x,x_1),\ldots,\sigma(x,x_k))^T, \end{aligned} \tag{3}$$

where R_k^{-1} is the inverse of the $k \times k$ covariance matrix of the random process at the locations where the function values are known.

The stochastic model has some very interesting properties. Firstly, the conditional mean $m_k(x)$ interpolates exactly at the observed points, and hence retains all information gained from exact measurements. Secondly, the variance $s_k^2(x)$ captures the uncertainity of predicting the value of the objective function at an untried $x \in A$. If the distribution of $\phi(x)$ is assumed Gaussian, then $m_k(x)$ and $s_k^2(x)$ fully specify the conditional distribution $\phi(x)$. This distribution can be used to guide the search for small values of $\zeta(x)$. It seems more likely to find a point with small function value where $m_k(x \mid .)$ is small. However, large values of $s_k^2(x \mid .)$ indicate regions of great uncertainity, i.e. regions where function values can differ greatly from the conditional mean. Hence a rational choice has to be discriminate between points of small mean but large variance or points of small variance but somewhat larger mean. We shall briefly review some proposed algorithms that fit the stochastic modeling paradigm.

2.1 Algorithms

Several algorithms for optimization using a stochastic model function have been investigated. We summarize some interesting approaches and finish with the P-Algorithm which forms the basis of our optimization procedure. These approaches essentially differ in the stochastic model chosen and the method used for minimizing the model function.

In Adachi et. al.[3], the model function is a stationary stochastic process. The conditional mean is an interpolating function. The optimal point is chosen by minimizing the interpolating function starting from the smallest data point found thus far, subject to a constraint on the coefficient of variation, which is the ratio of the mean and variance of the conditional distribution. The procedure is iterative, each iteration leading to (hopefully) a different local optimum of the objective function, so that, eventually, all local optima can be located. The auxiliary computations are expensive, namely the use of a nonlinear optimizer at every iteration to find the minimum of the interpolating function.

Bernardo et. al. [1], employ a stochastic model function to perform yield optimization of electronic circuits. Their approach relies heavily on designer intervention to identify significant parameters through parameter effect plots and also to identify promising sub-regions in the design space.

The P-algorithm was developed and characterized in [9]. It is an iterative procedure. At each iteration, a new observation point x_{k+1} is chosen that has the highest probability of $\phi(x_{k+1})$ being smaller than y_{ok}, i.e.,

$$x_{k+1} \quad = \quad \text{Arg } \max_{x \in A} P_x(y_{ok}) \tag{4}$$

is chosen as the next observation point where, y_{ok} is some value less than $min_{x \in A} m_k(x \mid x_i, \zeta(x_i), i = 1,\ldots,k)$, and

$$P_x(y_{ok}) \quad = \quad \text{Probability}(\phi(x) \leq y_{ok})\ . \tag{5}$$

The P-algorithm is a general formulation of a strategy to maximize the information gained by each function evaluation, and is quite easy to implement. We now detail our implementation of the P-algorithm. The process is iterative, hence the user can stop the iterations any time the best existing solution is satisfactory. The number of simulations to be run are directly controlled by the user. The algorithm only identifies the most promising points for simulation.

1. Choose k points x_i, $i = 1, \ldots, k$ uniformly from A using random sampling and compute $\zeta(x_i)$ by simulation. Start iteration $l = 1$.

2. Using the BLUP and MSE expression in [8], find the mean $m(x_j)$ and variance $s(x_j)$ at $N \gg k$ uniformly distributed points in A.

3. Find the smallest value of $m(x_j)$, i.e.

   $$m_l(x) \quad = \quad \min_{j \in 1 \ldots N} \; m(x_j) \; . \qquad (6)$$

   Let $y^l_{ok} = m_l(x) - \epsilon_l$. At each x_j find the probability $P_{x_j}(y^l_{ok})$.

4. Choose n_l points with largest probability $P_x(y^l_{ok})$ from the N points.

5. Compute $\zeta(x)$ at the n_l points found above. If $\min_{j \in 1 \ldots n_l} \zeta(x_j)$ is satisfactory, then stop, else continue

6. $k = k + n_l$. If $k > K_{max}$, then stop, else $l = l + 1$, go to step 2.

This algorithm is parameterized by the constants k, n_l, ϵ_l and K_{max}. These parameters have to be adapted to the specific problem or left to the designer's judgement. For example, $k = 10 * d$, $n_l = 2d$, where d is the dimensionality of the design space A, were found to be good values for problems below. In this way the designer directly controls the number of simulations to be run. The choice of N search points can be suitably biased by the designer's judgement, and can account for constraints on the design space, e.g. if the design region is a polytope constrained by linear inequalities on the design variables. Remember that only the mean and variance has to be estimated at the N points, which is quite inexpensive.

This leads us to another very important question, namely, handling constraints on the design. If the constraints are analytic, they can be handled very naturally by screening each of the N points for violation. When the constraints are implicit and can be checked for only after simulation, e.g. a maximum delay or power restriction when doing a skew optimization, the procedure needs to be modified. If the constraint can be evaluated through the same simulation, then another stochastic model can be used to model the constraint. A certain tradeoff has to be established between evaluation of this secondary model and the actual objective. If a penalty method is used to incorporate the true objective and the constraint in a single objective function, then the optimization task is a little more difficult, as the overall model might have to account for disparate variations. The formulation is flexible enough to allow for either option, namely, screening the evaluated points using the constraint model, or incorporating constraints directly into the objective function by penalty methods. The former approach is used in the first example given in the next section, where a delay controlled element has to be optimized for maximum delay variation, subject to a maximum delay constraint. The second, a clock driver circuit, is an essentially unconstrained optimization problem.

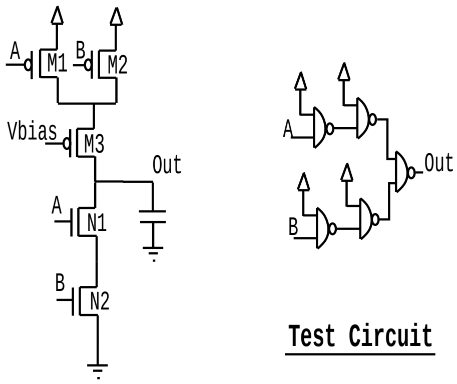

Figure 1: Circuit Block for Wave-pipelining

3 Optimization examples

3.1 Delay Controlled Element for Wave-pipelined circuits

The design of wave-pipelined circuits involves very careful control of the delay of each path in the combinational blocks. For static CMOS gates, the delay varies considerably with input data. A gate structure suitable for wave pipelining is shown in figure 1. Here the transistor M3 is used to add extra resistance to the pull-up chain to reduce the effect of the simultaneous switching of both inputs. It also has the deleterious effect of slowing down the circuit. Hence a proper balance has to be struck between the maximum delay through the gate, as well as the data-dependent spread [5]. The delay spread has to be minimized over process variations also. Hence the goal of the optimization is to obtain a suitable sizing scheme such that the delay spread through each circuit block is minimized, subject to a constraint on the maximum delay through the circuit.

The optimization problem is formalized as follows:

$$\begin{aligned} \text{Find } \boldsymbol{x}^* &= \text{Arg min}_{\boldsymbol{x}\in A} \; \text{max}_V \; \delta^*(\boldsymbol{x}) && (7) \\ \text{subject to} \quad & \text{max}_V \; \text{delay(x)} \leq D_{max} \; . && (8) \end{aligned}$$

Here V denotes the nominal and the four process corner MOSFET models. A is the hypercube formed by restricting the widths of M1-M3 between 3.6μm and 10μm, and V_{bias} between 0.0 and 2.0 V. The widths of N1 and N2 are constrained to be one-half the width of M1 and M2 respectively. $\boldsymbol{x}$ is the vector of feasible transistor widths and bias voltage. Note that the minimum allowed feature size is 0.6μm and hence the widths of N1 and N2 were restricted to vary in quanta of 0.6μm. The skew $\delta^*(\boldsymbol{x})$ is defined as the variation in delay through the circuit shown in figure 1 over the six possible input transitions, and the $delay(\boldsymbol{x})$ is the largest delay over these input transitions. D_{max} is the maximum delay constraint.

The models for delay and skew were initially established by simulating $k = 100$ different sizing schemes, selected randomly. The first row of table 1 shows the sizing scheme with the best skew value, satisfying the delay constraint, among these 100 points. This sizing scheme is not feasible.

Since the number of possible sizings is small, all the feasible alternatives (216 distinct sizing schemes) were evaluated with five values of bias voltage ranging from 0 to 2.0V using the BLUP and MSE expressions in [8]. This constitutes an exhaustive search of the design space using the models. Since the smallest possible value for the skew is 0, y_{ok} was chosen to be zero. 20 feasible sizes and bias voltages with the largest probability that satisfied the delay constraint $D_{max} \leq 1ns$ were chosen for resimulation. Table 1 shows the results for these sets of simulations. The best sizing in the second set was considerably better than the results of the first 100 samples and was considered quite suitable for design and hence no further simulations were performed. The total time taken for simulation was 540 cpu seconds on a DECstation 5000 while the overhead of model building and searching was less than 1 cpu second.

This example illustrates how the optimization procedure is employed. The search space is pruned by the designer's judgement and a good solution is found with very few simulations.

3.2 CMOS Clock Driver Circuit

The second example we present here is the skew optimization of a single to differential input clock driver circuit shown in figure 2. It is desired to obtain a signal and it's complement from this circuit's outputs such that there is minimum skew between the two signals, i.e. to minimize δ^* (see figure 2) which is the maximum of the high and low skew between the clock signal and its complement. Again, this skew has to be minimized over the process variations. This optimization has to be done using a suitable transistor sizing scheme. The absolute delay through this circuit is not a concern, hence the optimization is essentially unconstrained. Temperature and power supply variations were also considered. The model was built over the space of transistor sizes, process, temperature and power supply variations. As in the previous example, process variations were considered by simulating each sizing scheme over the 4 process corners and the nominal process.

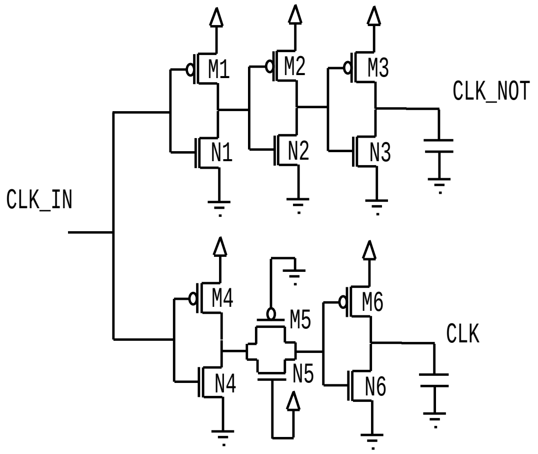

Figure 2: Clock Driver Circuit

The problem is formalized as follows:

$$\text{Find } \boldsymbol{x}^* = \text{Arg min}_{\boldsymbol{x}\in A_w} \; \text{max}_E ; \text{max}_V \; \delta^*(\boldsymbol{x}) \quad (9)$$

Here, A_w is the hypercube formed by restricting the widths P1-P6 between 3.6μm to 12μm, and E represents the temperature variation between 25-75° and V_{dd} between 4.75-5.25 V. As before the widths of N1-N6 are constrained to be one-half the widths of P1-P6 respectively. V represents the process variations considered. For this problem, the sizing provided by the resident circuit design expert had a worst case skew of 290ps (row 1 of table 2). For optimization purposes, the worst process dependent skew model was built using $k = 100$ simulations, selected randomly again. First, the effect of temperature and supply variation was factored out. To do this, 1000 random points were sampled in (A_w). At each of these 1000 points, the *model* was evaluated for 9 different combinations of the supply voltage and temperature variations. The largest value of the probability $P(y_{ok})$ (equation 5) over these 9 combinations was found for each of the 1000 points. This value was used to estimate

Table 1: Results for Delay Controlled Element

	M1	M2	M3	vbias	delay	skew
Best Random Sizing	6.0e-6	8.4e-6	8.4e-6	.98	1.0e-09	7.7e-10
Best Sizing after Optimization	7.2e-6	8.4e-6	9.6e-6	0.0	8.8e-10	5.4e-10

Table 2: Results for Clock Driver Circuit

	M1	M2	M3	M4	M5	M6	skew
Designer's choice	9.6e-6	4.8e-6	4.8e-6	9.6e-6	4.8e-6	4.8e-6	2.9e-10
Optimal Point	7.2e-6	3.6e-6	8.4e-6	9.6e-6	9.6e-6	4.8e-6	1.1e-10

the likelihood of a sizing being the best. i.e. :

$$x^* \quad = \quad \text{Arg } \min_{x \in A_w} \max_E P_{x_w}(y_{ok}) \qquad (10)$$

was the target for further simulation. Again y_{ok} was chosen to be 0.0 which is the minimum possible value of the skew. From the 1000 sizing schemes evaluated, the 10 sizing schemes (instead of only one as suggested by equation 10) with the largest probability were chosen for further simulation. These schemes were verified using the 5 process parameters and the 4 corners of power supply and temperature fluctuations. The smallest worst-case skew among these sizings was only 110 ps, a significant improvement over the expert's design (row 2 of table 2). The total simulation time was 640 cpu seconds on a DECstation 5000 and the overhead of model building and optimization was less than 10 cpu seconds.

These examples illustrate the power of this approach in evaluating transistor sizing schemes efficiently for different optimization problems. The stochastic model is able to capture the relationship of any performance measure to the transistor sizes. The model accuracy is reflected in the few subsequent evaluations required to arrive at very good designs.

4 Conclusion and Future Work

We have demonstrated an algorithm based on stochastic modeling that is capable of solving some difficult transistor sizing problems. This approach is very useful for general transistor sizing in CMOS circuits, where the objective function can only be evaluated through circuit simulation. We are currently exploring the effectiveness of this technique for problems with higher dimensionality, e.g. clock network design, and CMOS analog circuits [10].

References

[1] M. C. Bernardo et. al. Integrated circuit design optimization using a sequential strategy. *IEEE Transactions on CAD*, 11(3):pp. 353–360, March 1992.

[2] J. P. Fishburn and A. E. Dunlop. TILOS: a posynomial programming approach to transistor sizing. In *Proc. of ICCAD*, pp. 326–328, Nov. 1985.

[3] J.Q. Lu and T. Adachi. A parameter optimization method for electronic circuit design using stochastic model function. *Electronics and Communications in Japan*, 75(4):13–25, 1992.

[4] M. D. Matson and L. G. Glasser. Macromodeling and optimization of digital MOS VLSI circuits. *IEEE Transactions on CAD*, 5(4):pp. 659–679, Oct 1986.

[5] W. Van Noije, C. T. Gray, W. Liu, T. Hughes, and R. Cavin. CMOS sampler with 1GBits/s bandwidth with 25ps resolution. In *Proceedings of CICC*, pages 27.5.1–27.5.4, 1993.

[6] W. Nye, D. Riley, A. Sangiovanni-Vincentelli, and A. Tits. DELIGHT.SPICE: An Optimization-Ba sed System for the Design of Integrated Circuits. *IEEE Transactions on CAD*, 7(4):pp. 501–519, April 1988.

[7] E. S. Ochotta, R. A. Rutenbar, and L. R. Carley. Equation-free synthesis of high-performance analog circuits. In *Proc. of MIT/Brown Conference on Advanced Research in VLSI*, pages 129–143, 1992.

[8] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):pp. 409–435, 1989.

[9] A. Torn and A. Zilinskas. *Global Optimization.* Springer-Verlag, 1989.

[10] S. Mehrotra. *Simulation based design and optimization of digital circuits.* Ph.D. thesis, ECE Dept., North Carolina State Univ., 1994.