Analog Circuit Design Automation for Performance

Zhen-Qiu Ning, Marq Kole, Ton Mouthaan, and Hans Wallinga

MESA Research Institute
Dept. of E.E., Twente University
The Netherlands

Abstract

This paper describes an improved version of the program SEAS (a Simulated Evolution approach for Analog circuit Synthesis), in which an approach for selection of alternatives based on the evaluation of mutation values is developed, and design automation for high performance comparators is covered.

1. Introduction

The analog circuit design automation system, SEAS [1], is based on simulated evolution approach in which the current generation of the design is optimized with respect to the specifications and then modified for the next iteration. In this way the design generation is iteratively driven towards meeting the global specifications imposed by the designers, and resulting the "best" solution (selected from a list of feasible solutions). The synthesis is performed by an optimization algorithm based on simulated annealing to permit any kind of cost function and any starting point, and by a topology evolution system which finds out design problems and makes modification decisions of topology. Key to these modification decisions is a performance evaluator which examines the current synthesis and interactively reports its findings to the evolution system. The topology evolution system consists of the performance evaluator, a scoring technique, selection of alternatives, and topology reconstruction.

In the previous version of the program [1], the selection of alternatives is knowledge-based, and only opamp design automation is covered. The evolution direction much depends on the quality of the heuristic rules used. Opamps are circuits which can be described by linearized models. the models, however, may often not be applied to the field where many analog circuits exhibit non-linear behaviour. It is necessary to identify, model and manage non-linearities, transients, changes in operation region, etc., in high performance design automation [2].

This paper describes first how to extend the score technique to select the alternatives, then shows how design automation in SEAS is modified when an object with non-linear behaviour, for instance high performance comparator synthesis, is covered.

2. Simulated evolution process

The simulated evolution process in SEAS is shown in Fig.1. At the beginning of the process, a seed configuration, represented by a realized state vector and a sizing vector with starting point and reasonable upper bound as well as lower bound, is initialized. Then the sizing vector is optimized over all constraints. A state vector is a set of symbolic strings whose members represent the hierarchical breakdown of a design into sub-blocks. If the state vector has been assigned, it represents a definite circuit schematic. A sizing vector is defined as a minimal set of independent design parameters which determines the DC operating point of a circuit: bias currents and bias voltages. If the sizing vector is evaluated, it represents a definite sized transistor-level circuit diagram. After the optimization, the resulting sized circuit is under analysis: the design performance is measured in terms of the global constraints and quality calculations, and comparing the results of the design with the desired constraints is also carried out. If all the constraints are satisfied, we succeed in making a feasible solution. The feasible solution is stored and the

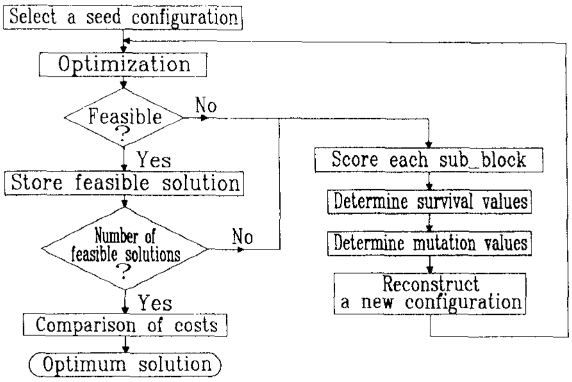
Select a seed configuration
Optimization
Feasible ?
No
Yes
Store feasible solution
Number of feasible solutions ?
No
Yes
Comparison of costs
Optimum solution
Score each sub_block
Determine survival values
Determine mutation values
Reconstruct a new configuration

Fig.1, The simulated evolution process in SEAS.

** *This research was support by the Innovative Research Program (IOP) on IC-technology under IC-TEL77.030.*

0-7803-0246-X/92 $3.00 © 1992 IEEE

number of feasible solutions required by users definition is examined. If a design is not feasible or the number of the feasible solutions is not satisfied, the design process will turn to the simulated evolution branch. To describe the simulated evolution branch, we review the following definitions. A score of a sub-block design style is a value which shows the mutation necessity of the sub-block design style in the current circuit (the current generation). A survival value of a sub-block design style indicates the mutation necessity of the sub-block design style over the next circuit (the next generation). A mutation value of branching from a design style to another design style of a sub-block is defined as their branching probability. With the survival value of each sub-block design style in hand, one can decide which sub-block design style has been failed to survive. Selection of an alternative is then based on a list of mutation values, if they are also at hand. Finally, a different configuration will be rebuilt at the end of the branch, and will become a new seed coming back to the beginning of the closed evolution loop.

Because of space limitation, we omit a complete description of all the simulated evolution steps, which appear in [1]. Here, we focus on improvements in the evolution loop: mutation value calculation and adaptive reconstruction technique.

A mutation value, say P_{kij}, represents a probability of branching from design style i to design style j of sub-block k under the influence of constraints (under the pressures of the environments). The probability is a weighted sum of sub-probabilities related to each performance. A sub-probability related to the lth performance, say p_{kijl}, can be determined by the knowledge-based approach [1], or calculated by a normalized difference of the lth performance between the current conditions (say design style i) and the preconditions (say design style j) of the same block (say block k). p_{kijl} has a value between -1 and +1. Typically, $p_{kijl}=-1$ represents the worst branching, $p_{kijl}=+1$ indicates the best change, and $p_{kijl}=0$ means no influence. The mutation value is used for guiding the search for alternatives and building up a different configuration rather than making a design decision, this makes its calculation more easier. The precondition information can be obtained from the calculation of starting point for optimization. If a sub-block has only two design styles available, it is not necessary to calculate the mutation values. Selection of alternative is simply the branching from the current design style to the other one.

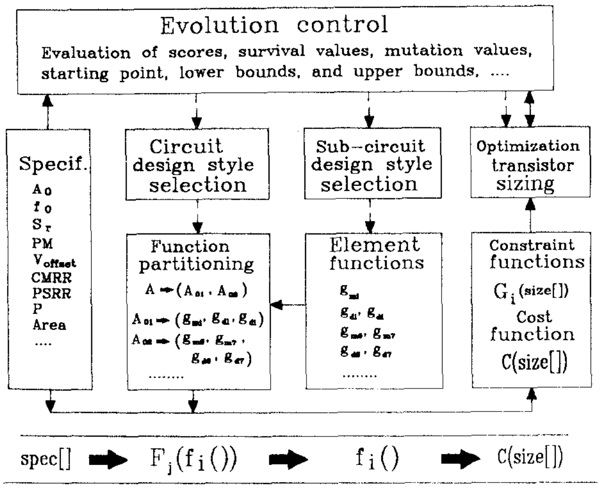

Fig.2, Topology evolution control and the corresponding cost function. The dashed line shows the control path, the solid line indicates the data path.

Once the current state vector is updated, the subsequent sizing vector, cost function, and constraint function of the new circuit have to be modified with the new alternative in replace of the original one. This is implemented by an adaptive reconstruction technique [1]. For op amps and comparators, important component values are widths and lengths of MOSFET's, and bias currents. These values are not all independent. If a general transistor model is used, they can be calculated by a sizing vector [1]. Performance of an op amp or a comparator is then a non-linear function of sizing vector through some element functions (see Fig.2). The element functions can be transconductances (g_m), output conductances (g_d), current factors (K), area (WL), and et al of certain transistors in a circuit, which are in turn functions of the sizing vector of the circuit. Key functions of the adaptive reconstruction technique are

- * to determine a sizing vector and element functions for an access state vector accordingly, and
- * to update cost function and constraint function for the next optimization.

In order to determine a sizing vector, a set of constraints, that link bias currents and bias voltages of all transistors in a circuit into groups whose members depend on each other, should be defined. This can be done by using expert design knowledge and first-order circuit analyses. The element functions are derived from a general transistor model. All the constraints and element functions are written into a database, and are accessible to the state vector as well as the sizing vector. After the state vector and sizing vector are updated, the cost function and constraint functions can be rebuilt by an adaptive function substitution in which the element functions corresponding to the replaced sub-block of the circuit are substituted by the element functions of the replacing sub-block. In the database, only the element functions are directly related to transistor model and technology. As the transistor model and/or the technology evolve, only the small part concerned need to be modified.

3. Synthesis of comparators

Key performance of a comparator is low-power, high-speed, and low-offset voltage. Offset cancelling technique is often used in an analog MOS comparator. This may cause, however, a large noise in the converted signal in a noisy environment, as in a flash A/D converter [3]. Difficulty for a comparator design automation system is how to build up an analytical model in which analytical design equations to express dependency of the comparator performance on design variables are well presented, and one can optimize the performance without using any offset cancelling technique based on the model. In order to show the model, we first describe design style, then present design constraints we considered. We show design example to support the model, and finally discuss some existing problems and the future directions.

3.1, Design style

A comparator design style is normally defined as an interconnection of abstract blocks. The comparator design style we chose is latch-based (see Fig.3). It consists of a differential stage, similar to the ones used in opamps' synthesis [1], but here in the differential pair the n-channel transistors are substituted by the p-channel transistors and a latch. The design style is hierarchical, and is completed by alternating selection of the design style of sub-blocks. For each sub-block there are at least two design styles available. Two design styles for the latch circuit are regenerative [2] and charge/discharge (see Fig.4). Adding a design style to SEAS requires specifying the topology of sub-blocks, and building a simulated evolution loop for refining global specifications down to the sizes of components for the sub-blocks. The difficulties are to develop an evolution control in which scores, survival values and mutation values are evaluated, and to build up a sub-database which consists of all constraints for design variables and element functions for comparator design equations.

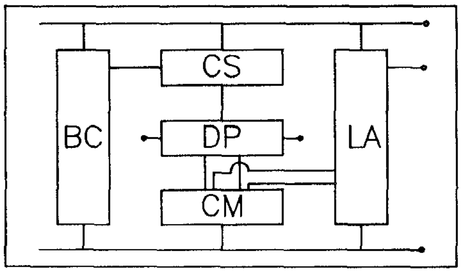

Fig.3 Design style of comparator.

3.2, Design constraints

The design constraints of the comparator we considered are listed in Table 1. The common mode input voltage range can be derived from the fact that each transistor in the input stage is worked in the saturation region. There are two kinds of offset voltage: systematic offset and mismatch offset. The first is due to a non-symmetric design which is not the case in our process. The mismatch offset is due to charge mismatch and dimension mismatch of the MOS transistors in the input stage. We assume that the feedthrough from the latch stage to the unbalanced output impedance of the input stage is negligible. Here we only deal with the mismatch offset. The propagation delay time is defined as the time required from the beginning of the strobe pulse rise to the time when one of the outputs is within 5% of the supply voltage (the response at the end of three time constants). It is dominated by the time domain response of the latch, since the delay of the input stage tends to be much less. The latch stage operates primarily as a large-signal circuit, but the time domain response is largely dependent on the small-signal behaviour [2]. If the switch resistances of transistors Q_{L3} and Q_{L4} in Fig.4 are negligible, for instance, a second order characteristic equation can be derived to describe the latch circuit. Then the time domain response can be approximated by a single pole of the result transfer function. The clock frequency is related to the propagation time and the time required to return the latch stage to the initial conditions. Here only its upper bound is specified and approximated. The quiescent power dissipation is defined as the amount of the consumed power when the input of the comparator is in equilibrium. Since the switch transistors in the latch are not always conducting current, the momentary power consumed will be integrated over the clock-period. The area estimates transistor size only, the area of interconnections is not taken into account.

In SEAS, as shown above, state of a design is completely determined by a realized state vector which represents a definite circuit schematic, and by an evaluated sizing vector which represents a definite sized transistor-level diagram of the circuit. All the constraints (or performance) shown can be treated as functions of state vector and sizing vector. One can emphasize any performance by optimization of the state vector and sizing vector via the closed simulated evolution loop. One can optimize the offset voltage, the power dissipation and the frequency response, for instance, to reach low-power, high speed and low-offset without using any offset cancelling technique.

If a state vector,

$$\begin{aligned} \textit{state}() = \textit{state}(&\textit{CIR=comp,DP=simple,CM=simple,} \\ &\textit{CS=simple,LA=charge/discharge}) \qquad (1) \end{aligned}$$

which represents the circuit schematic shown in Fig.4, is chosen, the offset voltage can be written as

$$\begin{aligned} V_{offset} = {} & \Delta(V_{T0})_{dp} \\ & + \sqrt{\frac{K_{cm1}}{K_{dp}}}\Delta(V_{T0})_{cm1} + \sqrt{\frac{K_{cm3}}{K_{dp}}}\Delta(V_{T0})_{cm3} \\ & + \frac{1}{2}\sqrt{\frac{AI_0}{K_{dp}}}(\delta K_{dp} + \delta K_{cm1} + \delta K_{cm3}); \qquad (2) \end{aligned}$$

with V_{T0} the threshold voltage when $V_{SB}=0$, K a current factor of transistor ($K = \mu\ C_{ox}\ W/L$), I_0 the tail current of the input stage, ΔV_{T0} a mismatch of threshold voltage, δK a relative mismatch of current factor, and A a modified factor for the transistor model used. The first three terms come from charge fluctuation, the last term comes from dimension fluctuation.

If we assume that the transistors in a pair, for instance the transistors Q_{dp1} and Q_{dp2} of the differential pair, are closely spaced, and that mismatch is estimated by $2^{1/2}\sigma(x)$, with $\sigma(x)$ a standard deviation of variable x, then the mismatches $\Delta(V_{T0})$ and δK can be calculated by the model developed in [4] as follows:

$$\Delta(V_{T0})_j = (A_{VT}\sqrt{\frac{2}{WL}})_j \qquad (3)$$

$$\delta K_j = (A_K\sqrt{\frac{2}{WL}})_j \qquad (4)$$

where j (= dp, cm, ...) denotes different transistor pairs, A_{VT} a area proportionality constant for threshold voltage, and A_K a area proportionality constant for current factor. The proportionality constants can be approximated in terms of the data published in [4], which are presented as a function of transistor gate oxide thickness. If the gate oxide thickness is 50 nm, for instance, A_{VTn} will be 30 (mVμm) and A_{Kn} 2.3 (%μm).

Since the current factors (K) in equ. (2) and the areas (WL) in equ. (3) and (4) are functions of the sizing vector related to the circuit shown in Fig.4, one can optimize the sizing vector to reach a low offset voltage for the given state vector (1) via a numerical optimization algorithm (the simulated annealing or the "Fletcher-Reeves" method). As the state vector can be reconstructed in the evolution loop if necessary, the comparator circuit schematic and its components are both optimized.

4. Results and discussions

The program was written in C. A design example is shown in Fig.4 and Table 1. There is no simulated result for the offset voltage in Table 1. This can be done by using SPICE Monte Carlo simulation. It is under study.

SEAS deals with the comparator design automation as an optimization problem of both circuit topology and its component values based on the simulated evolution. It can support complex performance requirements, and can adapt easily as transistor model and/or technology evolve. Some comparators have been designed and successfully simulated. The CPU time required is mostly expended on the numerical optimization. To reduce the CPU time, a modified simulated annealing algorithm is under development, in which variables are listed in terms of scores, and only important variables are updated. The mutation value for branching is calculated, only if more design styles for a sub-block is available. Now only the op amp output stage is used to check the idea. We will extend the comparator database to include more sub-block design styles [5], and complete the simulated evolution loop.

Table 1, Specifications and result performance for the example given in Fig.4.

Performance	Specified values	Predicted values (SEAS)	Simulated results (SPICE)
$(V_{in})_{min}$ (V)	< 1.25	0.88	0.43
$(V_{in})_{max}$ (V)	> 3.75	3.84	4.09
V_{offset} (mV)	< 3.0	2.3	no result
t_{prop} (nS)	< 10.0	2.48	2.63
f_{clock} (mHz)	> 25	< 58	< 103
P_{ower} (mW)	< 1.0	0.078	0.021
A_{rea} (squ.)	< 250	177	no result

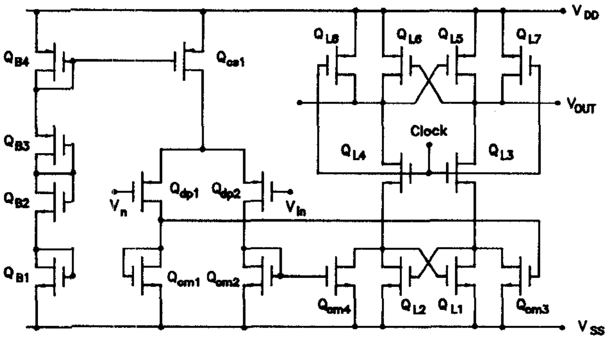

Fig.4, A comparator design example.

References

[1] Z.Q. Ning, T. Mouthaan, and H. Wallinga, "SEAS: a Simulated Evolution Approach for Analog Circuit Synthesis", IEEE CICC'91, San Diego, May, 1991
[2] R. Harjani, R.A. Rutenbar, and L.R. Carley, "Analog Circuit Synthesis for Performance in OASYS", in Proc. 1988 IEEE Int. Conf. CAD, Nov. 1988
[3] A. Yukawa, "A CMOS 8-Bit High-Speed A/D Converter IC", IEEE J. Solid-State Circuits, Vol. SC-20, No.3, June 1985
[4] M.J.M. Pelgrom, A.C.J. Duinmaijer, and A.P.G. Welbers, "Matching Properties of MOS Transistors", IEEE J. Solid-State Circuits, Vol.24, No.5, Oct. 1989
[5] M.E. Kole, "Comparator Design Automation in SEAS", MSc. thesis, Dept. of E.E., Twente University, Sept. 1991.