Performance Estimation in Analog Computer Aided Design

I. Faik Baskaya* and Günhan Dündar*

***Abstract* – In this paper, part of a joint effort in developing an analog design automation system with different levels of synthesis has been described. The study has been restricted to various types of analog CMOS circuits. Transistors have been modelled using analog neural networks trained with data obtained from simulation tools. Using these models in a circuit block, DC operating points have been determined iteratively and power dissipated by the circuit has been calculated in the DC analysis. In the AC analysis, gain and bandwidth of the system have been calculated, completing an evaluation cycle. Evaluation cycles have been used in a search mechanism where a range of input parameters have been swept in order to obtain the performance limits of the given circuit topology.**

1 Introduction

Automation in analog design is attracting increased interest because of the large amount of time and effort spent in manual design. As an efficient strategy for automation, hierarchical decomposition, which helps dividing the complete design task into sub-tasks may be proposed [1]. Typically at the top of the flow, system level synthesis, governed by system specifications, which decide the architecture to be used, takes place. Next level can be called circuit level synthesis, as it will be dealing with optimization of circuit blocks used in the macro-models of the higher level, taking device model parameters into account. Finally, at the lowest level will be layout synthesis, in order to give out the physical realization of the circuit incorporating process and fabrication data [2-4].

If; however, the circuit level synthesizer can not find a possible solution to the optimization problem of a certain circuit block demanded by higher level synthesizer; i.e., such a circuit is not possible, some changes will have to be made in the architecture, thus nullifying all the efforts at the circuit optimization level. In order to overcome such a problem, either the output parameters of the circuit have to be expressed in terms of each other so that a decision can be made analytically or a region has to be searched by sweeping the input parameters and evaluating their respective outputs. Since the analytical method is a tough one and the performance evaluators being used in many circuit optimizers are very slow for such a mission, an approximate albeit faster estimation of the performance would be helpful in order to search whether a desired solution exists in a certain region prior to further optimization efforts.

*Bogaziçi University, Department of Electrical and Electronic Engineering, Bebek 80815, Istanbul, Turkey, e-mail:dundar@boun.edu.tr, Tel: +90 (212) 263 1540 ext 1860, Fax: +90 (212) 287 2465

In this work, an approximate performance estimation tool for only four of the performance specifications; namely, gain, bandwidth, power consumption, and layout area of an analog block will be described. High accuracy is not demanded; however, high speed is a must. Using such a high-speed estimation, checking a large design space in an acceptable time will be possible in order to determine performance limits of the relevant circuit block. To this end, a fast performance estimator was developed to be used for generating circuit performance ranges. The estimator utilizes neural networks and/or equations to model the DC behavior and small signal parameters of transistors, an iterative methodology for estimating DC operating conditions, and an equation based approach for estimating AC performance parameters in an evaluation cycle. This estimator is run for many input values swept in a user-determined range to generate a data set from which circuit level performance limitations can be deduced. The overall system is depicted in Figure 1.

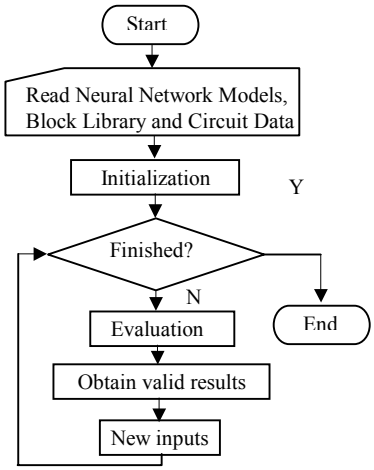

Figure 1: Flowchart of the performance estimation and design space exploration system.

In the next section, neural network based MOSFET modeling has been described. Sections 3 and 4 discuss the DC and AC analysis steps of the evaluation cycle. Section 5 explains search mechanism of the system. Section 6 concludes the paper.

2 MOSFET Modeling with Analog Neural Networks

A device model should be as accurate as demanded by the application and as simple as possible meanwhile [5]. SPICE is a tool which approaches MOSFETs with several models each with a different level of complexity, computational cost, and accuracy [6]. Although it is being used as an evaluator in the core of many circuit optimizers, it is more accurate than necessary and slower than acceptable for a fast performance estimator, which will have to estimate performance of many circuits in an input space region.

2.1 Obtaining Training Data and Modeling

If; however, enough data from SPICE simulations of a transistor can be collected, a neural network which will represent the operation of a single transistor can be trained and that model can be used instead of SPICE model of that transistor in evaluations. Neural networks have been used to solve problems of processing massive amounts of noisy and highly redundant information for a very long time in the nature: for about 600 million years [7]! There have been earlier attempts in training neural networks for modeling MOSFETs [8]. Manufacturing variations and mismatches in the parameters have also been studied in an earlier research [9]. In this study, multilayer perceptron networks have been trained using backpropagation.

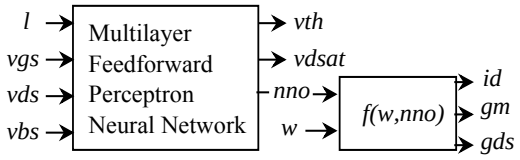

Figure 2. Transistor modelling using neural networks and neural network output.

Among the results of operating point analysis carried out by SPICE, *id*, *gm*, *gds*, *vth*, and *vdsat*, which will be called intermediate outputs, are necessary in order to calculate the DC operating point and carry out AC analysis of the circuit where capacitances which can be calculated easily using the geometry of the transistor (*w*,*l*) will be necessary. A transistor's operation will be determined by five parameters which define its terminal voltages and geometry: *vgs*, *vds*, *vbs*, *w*, and *l*. Since *w* has a more predictable effect on the intermediate outputs, number of neural network inputs can be reduced to four and the effect of *w* can be incorporated in a function taking the results of neural network outputs as input. Figure 2 depicts the approach for evaluating the intermediate outputs. For each type of transistor (N or P) manufactured with a particular technology, an independent neural network has to be trained. Also, in order that the networks can be better trained, an independent network for each of five necessary intermediate outputs can be preferred.

In order to train and use neural networks, data, which may have different ranges for each output, have to be mapped to a region (0:1). 0 and 1 are not included for preventing weights of the synapses from taking large values. If n is within (0:0.5), real values are mapped into a range [n:1-n] with the function:

$$y = \frac{x - x_{min}}{x_{max} - x_{min}}(1 - 2n) + n \qquad (1)$$

where x denotes real values and y denotes values in the range [n:1-n] to be used in neural networks. Since data of some of the outputs, *id* for instance, may have a wide dynamic range, and logarithmic mapping results in a very high error after mapping the results back, networks have been divided in order to learn different regions of that particular intermediate output's values. For that reason, 0.1 and 0.9 values are reserved to indicating whether the result for a given input set belongs to a lower region or an upper region. With a safe distance from those boundaries, data for that region is mapped to [0.25:0.75] during this study.

Patterns covering all possible combinations of each of four inputs of the network are obtained from SPICE simulations. Some of these patterns may be eliminated since they never occur using reasonable operating voltages. Since *gm* and *gds* will be used only if the transistor is in saturation, even more patterns will be eliminated for training networks involving them.

All these considerations result in several networks for each type of transistor. These networks can be trained using any training environment and a forward passing utility, which can read the model files of trained networks and may be invoked from within the estimation software.

2.2 Results of Training

For each of the PMOS and NMOS transistors, *vth* and *vdsat* were trained without dividing the patterns. For *gm* and *gds*, patterns were divided into two from the boundary 50 μS, and for *id*, they were divided into four at the boundaries, 10 μA, 100 μA, and 1 mA since *id* has patterns from all operating regions of the transistor. For *vth* and *vdsat*, networks with one hidden layer, and for the rest, networks with two hidden layers were used.

Interestingly, when the results obtained at the output of a network is converted back to its original range, error of the converted result is observed to be higher than that of the result at neural network output. Following from (1):

$$x + \Delta x = (y + \Delta y)\frac{x_{max} - x_{min}}{1-2n} + \frac{(1-n)x_{min} - n x_{max}}{1-2n} \quad (2)$$

$$\frac{x}{\Delta x} = \frac{y}{\Delta y} + \frac{1}{\Delta y}\left(\frac{1-2n}{x_{max}/x_{min} - 1} - n\right) \quad (3)$$

With a typical value of y=0.5 and a 10 % error at neural network output; i.e., Δy=0.05, the family of curves showing the relation of n and x_{max}/x_{min} to the error of real output values can be observed from Figure3. In our study, since n=0.25 was used, a 10% error at the output of the neural network is reflected as 20 % error to the real values.

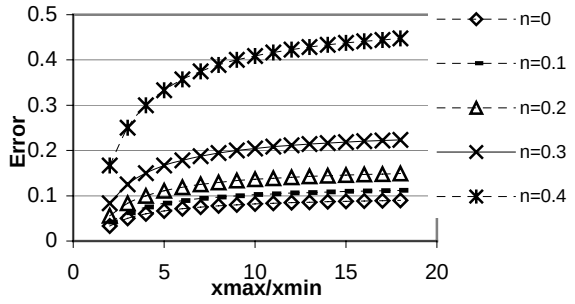

Figure 3. Reflection of a 10 per cent error in neural network outputs to the error in real values of outputs.

3 DC Analysis

After obtaining a valid model, DC analysis must be performed in order to determine the operating point, which is an equilibrium point of the circuit, forced by transistor sizes and bias voltages, and determines small signal parameters of the transistors. Hence, being a prelude to other types of analyses, this step is the core of the whole system [10]. However, the equations to be solved are usually non-linear algebraic systems, which cannot be solved directly. SPICE employs Newton-Raphson algorithm, which converts a non-linear equation to a sequence of linear equations, and solves them iteratively.

In order to get an approximation to the accurate solution on the other hand, some simplifications can be made. First of all, analog circuits will usually be a combination of certain analog blocks and in most of these blocks, transistors may be considered as sub-blocks connected in series between supply voltages and connected to each other through their drains or sources resulting in their sharing the same drain-to-source current. In a few blocks only, different situations such as branching of the current between supply voltages into two or more parallel paths may be dealt with exceptionally. Hence, a configuration for various analog blocks may be prepared and then used for calculating operating points.

3.1 DC Operating Point Calculation

In the previous section, the functions that will provide us with the intermediate outputs (*id*, *gm*, *gds*, *vth*, *vdsat*) given five inputs (*vgs*, *vds, vbs, w*, *l*) were defined. During an evaluation cycle, geometry of all transistors will be fixed; hence, *id* of the transistor will vary only by node voltages of the transistor and there may be only one operating point where node voltages and *id* of the transistors in a series path will be in equilibrium. Gates of the transistors will either be connected to external bias voltages or to a source or drain terminal of another transistor within the circuit. This gives us the chance to vary *id* of transistors by only varying their drain and source voltages, which will be ordered in a line between supply voltages.

On a column of transistors, one node close to either of the power rails is assigned to a voltage value and the corresponding currents is calculated. That current value is used to calculate the voltage of the other transistors on the column. If currents of all transistors don't match, voltage of the first node is assigned to a new value and this procedure is applied until the desired matching occurs. Updating node voltages in an adaptive manner helps faster convergence.

3.2 Completing DC Analysis

Having obtained operating point node voltages, remaining intermediate outputs may be calculated in a straightforward manner. Among these, *vth* will be used to check whether the transistor is in conduction or at cut-off and *vdsat* for checking whether it is operating in saturation region or not. Operation in saturation region is requested for all transistors, otherwise, the design will not be of use and hence, calculation of *gm* and *gds* will not be necessary. *gm* and *gds* values are calculated only if all the transistors within the block are at saturation.

Also, since *id* of a transistor is equal to the current flowing through the line, current supplied by the supply voltages is known and calculation of power is just a matter of multiplying difference of the supply voltages with this current.

There are some exceptions to this general routine such as the case where a path branches into two or more paths because of parallel connected transistors. In such situations, those transistors share the current flowing through the line by a ratio. Currently, only

the case where these parallel transistors are matched as in differential amplifiers have been researched; other situations may be further studied.

4 AC Analysis

In AC analysis, capacitance values must also be known. These values will be calculated from some of the process parameters and sizes of the transistors. A generalized AC analysis requires a symbolic analysis tool. The AC analysis equations have been obtained through hand analysis. Each block has equations for transconductance, output conductance, input and output capacitances of its own; each of these parameters of the overall block take their values from different transistors in different types of blocks. Once these parameters are known for all blocks that are cascaded in a circuit, overall gain and bandwidth of the circuit can be determined easily.

5 Searching Performance Limits

Evaluation of many circuits is necessary in order to determine performance limits of the given circuit. This will be established by incrementing all input values of transistor sizes and bias voltages from an initial value to a final value by an increment. After each increment, the circuit is re-evaluated and output values; i.e., gain, bandwidth, power, and area of the circuit, will be compared with performance criteria, which are the requirements ordered by the higher stage. Finally, all designs fitting the requirements will be listed and whether a desired solution exists or not may be seen easily.

6 Conclusions

In this study, an approximate but fast performance estimation tool was developed. This tool can explore large design spaces and find performance limits for CMOS analog circuits in a shorter time than widely used general-purpose circuit simulation tools. The purpose of developing this performance estimator is to aid other systems (namely, system synthesizers and circuit optimizers) with which it will be used in conjunction in an analog design automation system.

The entire work consists of different levels of analysis and estimation. With a bottom-up approach, first a transistor was modelled with analog neural networks which will obtain the necessary parameters of the transistor determined by its size and terminal voltages. With use of neural networks instead of iterative calculations of implicit equations, a boost in speed was realized; however, training of the networks need to be improved for better accuracy. For the time being, support from analytic expressions was provided in aid to the neural network models of the transistors in the calculation of *id*.

Results of DC and AC analyses have been compared to HSPICE results for simple analog blocks and minor differences between the two have been observed, which probably will be reduced with better modelling of transistors. For a preliminary test, a design space including 1619904 different designs was explored in 157 minutes; i.e., simulation of each design lasted for about 5.81 ms in comparison to HSPICE, which takes 48 ms per design with 10 points per decade and 240 ms per design with 100 points per decade. At the end of the search, 8087 designs obeying the design criteria were obtained. The tool allows choice among these designs.

References

[1] M. Ismail and T. Fiez, *Analog VLSI, Signal and Information Processing*, McGraw-Hill, Singapore, 1994..

[2] Erten, G., High Level Synthesis of Analog Integrated Circuits using Macromodels and Equation Based Optimization Tools, M.S. Thesis, Boğaziçi University, 1999.

[3] Alpaydın, G., A New Approach to Analog Integrated Circuit Optimization, Ph.D. Thesis, Boğaziçi University, 2000.

[4] Şimşek, A., Analog Layout Generation for Silicon Compilation, M.S. Thesis, Boğaziçi University, 1997.

[5] Engl, W. L., H. K. Dirks and B. Meinerzhagen, "Device Modeling", *Proceedings of the IEEE*, Vol. 71, No.1, pp. 10-31, January 1983.

[6] Antognetti, P. and G. Massobrio, Semiconductor Device Modeling with SPICE, McGraw-Hill, Singapore, 1988.

[7] Hertz, J., A. Krogh and R. G. Palmer, Introduction to the Theory of Neural Computation, Addison Wesley Publishing Company, California, 1991

[8] Ojala, P., J. Saarinen, P. Elo and K. Kaski, "Novel Technology Independent Neural Network Approach on Device Modeling Interface", *IEE Proc. On Circuits Devices Syst.*, Vol. 142, No.1, pp. 74-82, February 1995.

[9] Coşgül, G., A. S. Öğrenci and G. Dündar, "Neural Network Based CAD Tool for Modeling Manufacturing Variations in MOS devices", Proceedings of the TAINN, İstanbul-Turkey, 1999, pp. 202-209.

[10] Chua, L. O. and P. M. Lin, Computer-Aided Analysis of Electronic Circuits, Prentice-Hall, New Jersey, 1975.