

Two Level Performance Estimator for High Level Synthesis of Analog Integrated Circuits with Feedback

Adrián Núñez-Aldana and Ranga Vemuri

Digital Design Environments Laboratory
Department of ECECS, University of Cincinnati
Cincinnati, OH 45221-0030
{anunez,ranga}@ececs.uc.edu

Abstract

In this paper, we present a technique for estimating the gain, bandwidth, power and area of analog integrated circuits. A two-step approach is adopted to speed up the estimation process and handle larger analog systems. In the first step, the performance of basic analog components is estimated using a knowledge-based approach. Then, component models are generated with the estimates produced at this phase. In the second phase, we use a symbolic analysis methodology to evaluate the performance at the system level. The system net-list is represented by a signal flow graph (SFG) using the component models generated at the previous phase. The SFG approach allows to handle feedback loops at the system level.

1 Introduction

The analog synthesis process consists of two phases: topology selection and circuit sizing [1]. During the topology selection process, a net-list of components selected from a pre-defined library is generated. The generated topology should meet the specification requirements and possibly meet the system constraints. The circuit sizing phase evaluates the operating points and produces physical dimensions of all active devices such that the system constraints are met. Both phases, the topology generation and the circuit sizing, should be aware of the functionality and behavior of the components to be selected and sized. Typically, this knowledge is obtained from the experience of the analog designer, and through the use of circuit simulators.

In the development of a computer aided tool for synthesis of analog systems, the knowledge has to be embedded in a performance estimator. As the estimator is usually embedded in a searching engine, it is required to be fast and accurate. Having a performance estimator with high level of accuracy helps synthesis tools to make better choices at early stages of the design process. A fast estimator permits to

explore a bigger solution space in reasonable time, and it speeds up the convergence.

Feedback techniques [2] are very useful to improve the performance of analog circuits. In general, feedback alleviates the non-ideal characteristics of analog circuits. However, this technique can mask the effects of the low-level performance parameters at the system level, preventing searching engines to correctly allocate constraints. Therefore, a synthesis tool requires estimates at the system level as well as estimates of individual components without feedback.

In this paper, we present a methodology to estimate gain, bandwidth, power and silicon area of analog integrated circuits at the system level allowing feedback loops. A two-level approach allows us to estimate the performance of low level components and system level with feedback. The paper is organized as follows. Section 2 presents the motivation and related work. Section 3 describes our methodology. In section 4, we present results to show the accuracy and speed of the estimator. Finally, the conclusions are presented in section 5.

2 Motivation and Related Work

Two different techniques have been developed to evaluate the performance of analog circuits by academy and industry: knowledge-based and optimization-based [3]. The knowledge-based approach encodes the circuit behavior in memory, while the optimization-based obtains the behavior via simulation. The optimization-based methodologies (e.g. SPICE[4], AWE[5]) provide performance parameters at the system level, and they have the advantage of being applicable to all classes of circuits. However, they do not provide information of how each element affects the overall performance of the system, and they are computational expensive. On the other hand, the knowledge-based approaches are faster and provide more insight into the circuit.

```

procedure TwoLevelEstimator
  Modules  $\leftarrow$   $\emptyset$ ;
  Models  $\leftarrow$   $\emptyset$ ;
  NetList  $\leftarrow$  read(Circuit);
  for each component in NetList do
    Modules  $\leftarrow$  Modules  $\cup$  component;
  end for;
  while Modules  $\neq$   $\emptyset$  do
    component  $\leftarrow$  GetComponent(Modules);
    Modules  $\leftarrow$  Modules  $\setminus$  component;
    CompPerf  $\leftarrow$  APE(component);
    Models  $\leftarrow$  Models  $\cup$  CreateModel(CompPerf);
  end while;
  NewNetList  $\leftarrow$  ReplaceWithModels(NetList,Models);
  SystemPerformance := EstimateSFG(NewNetList);
  return SystemPerformance;
end procedure

```

Figure 1: Two Level Performance Estimator

But, they require the effort of analog designers to encode the performance equations, which limit them to a small subset of analog components.

Symbolic analysis [6, 7] techniques can be used to automatically produce analytic expressions of electrical systems. Tree enumeration [8], signal-flow graph [9], numerical interpolation [10] and parameter extraction methods [11] are some of the most general techniques that have been developed to produce symbolic expressions. Each methodology has advantages and disadvantages; but, in general, they have the drawback of exponential increase in their execution time and memory requirements as the number of nodes increases.

The methodology presented in this paper combines the fast evaluation technique of the knowledge-based approach and the automated characteristics of symbolic analysis.

3 Methodology

The methodology presented in this paper is structured in two levels. The first level uses a knowledge-based approach to estimate the performance of transistors, basic analog circuits (current mirror, differential amplifier, gain stage, etc.), operational amplifiers and components from an analog library. The second phase uses a symbolic analysis approach which transforms the system level net-list into a signal flow graph.

Figure 1 depicts the algorithm of the two level estimator. Two sets are used to carry the analog components information: *Modules* and *Models*. The *Modules* set contains all the basic analog components used in the system net-list. Each module is estimated using APE, a tool described in section 3.1. Using these estimates, the *Models* set is generated. Each model is built using RC circuits and controlled

sources, and it carries the individual performance parameters (area, power, etc.). Figure 2 shows a model of an operational amplifier generated after being estimated by APE. Then, *NewNetList* is created substituting the components in the original circuit by the respective models. Finally, the signal flow graph representation is constructed and estimated using a symbolic analysis method which is described in section 3.2.

3.1 Analog Performance Estimator

For the first level in the structure, we use an Analog Performance Estimator (APE) [12] developed at the University of Cincinnati, which is itself structured hierarchically. APE can estimate the performance of analog circuits for a given topology and fabrication process parameters. The estimation is performed in a bottom up fashion from the transistor level up to the component level. At each level in the hierarchy, the current design parameters are obtained from the parent and are further decomposed into the parameters for the sub-components. The estimation at different levels in the APE is described in the next paragraphs.

- **CMOS transistor level** : is the lowest level in the hierarchy of APE. The transistor is sized based on its DC operating point and the fabrication process parameters. In a CMOS transistor level, the parameters specify the electrical and process characteristics of the devices. Using these parameters and the transistor SPICE models, the large signal and small signal models of a CMOS transistor, capacitor and diode can be evaluated.
- **Basic Circuit Level** : The elements in this level include DC-bias voltages, current sources, gain amplifiers, output buffers, differential amplifiers, and differential-single ended converters. This level contains several topologies for each component, e.g. a current source can be implemented as a Cascode or a Wilson topology. APE contains a set of symbolic equations which relate the performance of the components to the circuit topology. The small signal characteristics of the transistors and the symbolic equations of the circuit are used to estimate the performance parameters of this circuit. For

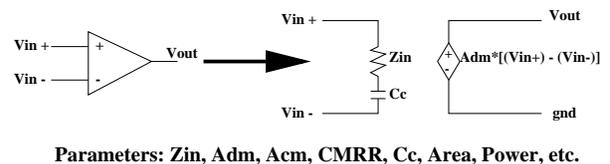


Figure 2: Operational Amplifier Model

instance, the typical performance parameters estimated for a differential amplifier are A_{dm} , CMRR, gate area, UGF, Zout, DC power and slew rate.

- **Operational Amplifiers :** This level consists of topologies of operational amplifiers. Each stage in the operational amplifier is composed of basic circuit elements. The performance of the operational amplifier is evaluated using the attributes of these basic circuit elements. For example, the UGF of the op-amp is computed as a function of gain and UGF of the differential amplifier and the gain stage, and the compensator capacitor.
- **Analog Module Level :** This library of analog modules forms the fourth level in the APE. Each of the modules are built using the operational amplifiers, elements from the basic circuit level, transistors, resistors and capacitors. This level consists of elements like inverting/non-inverting amplifiers, integrators, adders, sample-and-hold circuits, A-D, D-A converters, etc. The performance of these components is estimated using the operational amplifier attributes and the equations relating the ideal behavior of the component with the non-ideal characteristics of the operational amplifier.

Through several examples, APE has been proven to be fast and accurate. Components at the analog module level are estimated in tenths of seconds with maximum estimation error of 10% when compared with SPICE. This makes APE suitable for use within our system level estimator.

As APE encodes the analog expressions in memory, it is very efficient for estimating pre-defined analog components. However, it is not feasible to cover all kinds of analog components by APE. Therefore, we are limiting APE to basic analog components up to the level of operational amplifiers.

3.2 Signal-Flow Graph Estimator

The second level in the structure uses a symbolic analysis based on the signal-flow graph method. The signal-flow graph method, introduced by Mason in [9], was initially developed for the study of control systems; then it was extended to analyze electrical systems. The method consists in representing a system of differential algebraic equations as a SFG and evaluating the Mason's formula.

Applying the SFG method to general circuits has been proven to be inefficient for medium to large circuits. It requires to enumerate all directed paths and directed loops in the signal-flow graph, which both are NP-complete problems [13]. If the methodology is applied to a flat net-list, where the bottom

```

procedure EstimateSFG(NetList)
  ControlledSources  $\leftarrow$  GetSources(NetList);
  NetList  $\leftarrow$  RemoveControlledSources(NetList);
  tree  $\leftarrow$  GenerateTree(NetList);
  cotree  $\leftarrow$  GenerateCotree(NetList);
  VCotree  $\leftarrow$  ApplyKVL(tree);
  ITree  $\leftarrow$  ApplyKCL(cotree);
  for each branch in tree do
    VBrch  $\leftarrow$  branchImp * ITree[branch];
  end for;
  for each branch in cotree do
    IBrch  $\leftarrow$  branchAdm * VCotree[branch];
  end for;
  SFG  $\leftarrow$  crSFG(tree,cotree,VCotree,ITree,IBrch,VBrch);
  SFG  $\leftarrow$  SFG + ControlledSources;
  Num := 0;
  for each path in SFG do
    for each loop in path do
      Num := Num + path * loop;
    end for;
  end for;
  Den := 1;
  sign := -1;
  for each n-loop in SFG do
    Den := Den + sign*n-loop;
    sign := -sign;
  end for;
  estimate := Num / Den;
  return estimate;
end procedure

```

Figure 3: Signal-flow graph Estimator Algorithm

level elements are included (increasing the sizes of the flow graph), it becomes impractical for medium to large systems. Therefore, we substitute analog components (elements without feedback) by models generated by APE. In this manner, we can reduce the complexity of the signal-flow graph estimator.

Figure 3 depicts the algorithm which performs the estimation at the signal-flow graph level. The first step is to generate the signal-flow graph from the input net-list. A SFG is a weighted directed graph representing a system of simultaneous linear equations. The SFG is constructed from the Kirchoff voltage law (KVL), Kirchoff current law (KCL) and the branch voltage-current relationship (BVCR); therefore, it is considered a topological method. The net-list is divided into a tree and a cotree, such that the tree contains all nodes of the net-list without loops, and the cotree contains the complement of the tree. A tree-listing algorithm is used to enumerate the possible trees (cotrees) [14]. Using KCL and KVL, all the relationships between the tree and the cotree are evaluated, such that every node in the tree is expressed in terms of the cotree nodes, and every node in the cotree is expressed in terms of the tree nodes. Using the BVCR relationships each impedance and each admittance of the net-list are expressed in terms of the tree and cotree nodes. Once that the SFG is generated, the Mason formula

ckt	Estimates				Simulation			
	Area	Power	UGF	Adm	Area	Power	UGF	Adm
OpAmp	22080 μ^2	0.33mW	889.8KHz	1029.2	22080 μ^2	0.3mW	815.2KHz	942.3
InvAmp	2713 μ^2	0.89mW	88.98KHz	10.0	2713 μ^2	1.03mW	79.4KHz	10.06
Integrator	219000 μ^2	3.4mW	15.92kHz	1000	219000 μ^2	3.3mW	15.8KHz	941.2
Filter	659713 μ^2	11.09mW	966hz	1.0	659713 μ^2	10.93mW	977hz	0.93

Table 1: Estimates vs. simulation results

can be applied to estimate the transfer function:

$$T_{js} = \frac{X_j}{X_s} = \frac{\sum P_k \Delta_k}{\Delta}$$

$$\Delta = 1 - (\text{sum of all loop weights})$$

$$+ (\text{sum of all second-order loop weights})$$

$$- (\text{sum of all third-order loop weights}) + \dots$$

The Mason formula requires to enumerate the first order and the n th-order loops. To reduce the complexity of the algorithms, simple reductions in the signal-flow graph are applied first. The simplifications are described in [15]. Then, an enumeration algorithm [16] is used which first identifies the first order loops, then the n th-order loops.

The use of a SFG method allows to completely automate the generation of the transfer function of an arbitrary analog system. As the SFG includes only the component models, it can handle medium to large circuits in reasonable time.

4 Experimental Results

To verify our methodology, we applied it to several examples. We estimated an operational amplifier, an inverting amplifier, an integrator, and a second order low pass filter. In all cases, the operational amplifiers were sized and estimated first using APE. Then, models of the op-amp's were generated and used for the second phase. Each operational amplifier was specified as follows: (1) CMOS two-stage, (2) Wilson current source, (3) bias current of 10mA, (4) compensator capacitor 10pF, (5) differential gain 1000, (6) unity gain frequency 900Mhz. Table 1 shows the estimation and simulation results for the four examples, where the estimates were within a maximum error of 10%. The execution time to estimate all the components, including the generation of models, was less than 1 sec; while, the execution time of SPICE for all components was 20.2 seconds. Besides, a human interaction was needed to extract the performance parameters from SPICE, that was not necessary in our estimator.

5 Conclusions

An existing performance estimator for analog systems, APE, was extended to handle systems with

feedback. The extension was based on using a symbolic analysis technique, a Signal-Flow Graph method. The SFG methodology helped us to automate and estimate generic analog systems with feedback. The extension is being implemented to be part of a high level synthesis tool of analog systems.

Acknowledgment. This work was sponsored by the USAF, Air Force Research Laboratory, WPAFB under contract number F33615-96-C-1911. The authors thank Nazanin Mansouri for her help with the manuscript. Also, A. Núñez-Aldana would extend his gratitude to CONACYT-Fulbright for their support.

References

- [1] B. Antao, A. Brodersen, "Techniques for Synthesis of Analog Integrated Circuits", IEEE Design & Test of Computers, pp 8-18, Mar 92.
- [2] F. D. Waldhauer, "Feedback", New York, Chichester, Wiley, 1982.
- [3] L. R. Carley, G.E. Gielen, R. A. Rutenbar, W. Sansen, "Synthesis Tools for Mixed-Signal ICs: Progress on Frontend and Backend Strategies", DAC Proceedings 1996, pp 298-303, June 1996.
- [4] L.W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits", Memo ERL-M520, Berkeley, Calif, May 9, 1975.
- [5] L. Pillage, R. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis", IEEE Trans. CAD, vol 9. No. 4, pp. 352-366 Apr 1990.
- [6] P.M. Lin, "Symbolic Network Analysis", Elsevier, 1991.
- [7] F. Fernandez, A. Rodriguez-Vazquez, J. Huertas, G. Gielen, "Symbolic Analysis Techniques", IEEE Press, 1998.
- [8] R. M. Pederson, "User's manual for the ANP3 circuit analysis program", Lab. Circuit Theory, Tech. Univ Demark, 1972.
- [9] S.J. Mason, "Feedback Theory - some properties of signal flow graphs", proc. IRE (1953), 1144-1156.
- [10] J. Vlach, K. Singhal, "Computer Methods for Circuit Analysis and Design", Van Nostrand Reinhold, New York, 1983.
- [11] K. Singhal, J. Vlach, "Symbolic analysis of analog and digital circuits", IEEE Trans. Circuit Theory, pp 598-609, 1997.
- [12] A. Nunez and R. Vemuri, "An Analog Performance Estimator for Improving the Effectiveness of CMOS Analog Systems Circuit Synthesis", Conference Proceeding DATE'99, March'99.
- [13] N. Deo, "Graph Theory with Applications to Engineering and Computer Science", Prentice-Hall, 1974.
- [14] S. P. Chan, "Graph Theory and Applications", 1982.
- [15] H. Okrent, L. McNamee, "NASAP-70 user's and programmer's manual", UCLA, Tech Rept. ENG-7044, 1970.
- [16] P. Lin, G. Alderson, "SNAP-A computer program for generating symbolic network functions", Purdue University, TR-EE70-16, 1970.