Fast, Layout-Inclusive Analog Circuit Synthesis using Pre-Compiled Parasitic-Aware Symbolic Performance Models*

Mukesh Ranjan†, Wim Verhaegen§, Anuradha Agarwal†, Hemanth Sampath†,
Ranga Vemuri†, Geoges Gielen§
†Dept. of ECECS, University of Cincinnati, Cincinnati, OH-45221 USA.
§Dept. of EE, ESAT-MICAS, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium.

Abstract

We present a new methodology for fast analog circuit synthesis, based on the use of parameterized layout generators and symbolic performance models (SPMs) in the synthesis loop. Fast layout generation is achieved by using efficient parameterized procedural layout generators. Fast performance estimation is achieved by using pre-compiled SPMs, stored as efficient DDD-like structures called Element Coefficient Diagrams. Techniques have been developed to include layout geometry effects in the SPMs. The accuracy and efficiency of the parasitic inclusion technique as well as the proposed methodology have been demonstrated by comparisons to traditional synthesis methods. The proposed methodology is used for the synthesis of opamps and filters and is demonstrated to achieve effective performance closure.

1. Introduction

Analog circuit synthesis is the process of determining numerical values for the unsized circuit elements of a fixed circuit topology while satisfying a set of performance constraints. In a traditional circuit sizing [1] approach, a combinatorial optimization algorithm generates several alternative sets of component sizes. Each set of sizes is used for performance estimation by a numerical circuit simulator. The process converges when all constraints are satisfied. This process, however, suffers from two significant shortcomings. The performance of an analog circuit is sensitive to the parasitic effects introduced during the subsequent layout phase. This may lead to the failure of the circuit which is optimized without taking layout effects into account. The other drawback is the computationally expensive performance estimation due to numerical simulations in the synthesis loop.

The problem of performance degradation due to layout parasitics can be alleviated by either layout-aware or layout-inclusive circuit synthesis. While layout-aware techniques offer fast timing closure, the parasitics are difficult, and often impossible, to estimate accurately [2]. In layout-inclusive synthesis, layout generation and extraction is done within the synthesis loop. This captures the parasitic effects accurately [3].

*Sponsored in part by DARPA/MTO NeoCAD program under grant number F333615-01-C-1977 monitored by the Air Force Research Laboratory.

1530-1591/04 $20.00 (c) 2004 IEEE

Computationally intensive numerical simulations can be avoided by using circuit models for performance estimates. Accurate posynomial models are difficult to generate automatically [4] and are often created by expert designers [5]. Generation of neural network models is time intensive and is limited to a few circuit variables [6]. The generation and evaluation of accurate symbolic performance models (SPMs) is fast [7] and therefore is the method of our choice.

We propose a method which combines parameterized layout generators with SPMs for fast performance closure during analog circuit sizing. To combine the two aspects, algorithms for inclusion of layout effects into the SPMs have been presented and their efficiency and accuracy demonstrated. The paper is organized as follows. The proposed method is described in Section 2 followed by techniques to include layout effects in the SPMs in Section 3. Experimental results are presented in Section 4 followed by conclusions in Section 5.

2. Proposed Circuit Synthesis Approach

The proposed circuit synthesis environment is shown in Figure 1. Layouts are generated by using the Module Specification Language(MSL) system [8], which produces parameterized layouts. A parameterized layout is a fixed template layout, which when provided with the values of the circuit *parameters* by the optimization engine, produces a physical layout. In our case simulated annealing is used for optimization. A standard circuit extractor is used to extract the devices and parasitics from the layout. The extracted parasitic values along with the passive component values are passed to the pre-compiled SPMs. The SPMs also take in the small-signal parameter values for all active devices obtained by performing an operating point analysis using SPICE. The performance estimates obtained from SPMs are compared to the specified constraints. If necessary, the optimization engine proposes a new set of design parameter and this process continues till convergence.

2.1. Layout Generation and Instantiation

The MSL system is used to generate parameterized layouts. MSL contains constructs for hierarchical instantiation and relative placement of modules and for defining parameterized nets for routing. When compiled an MSL program yields an

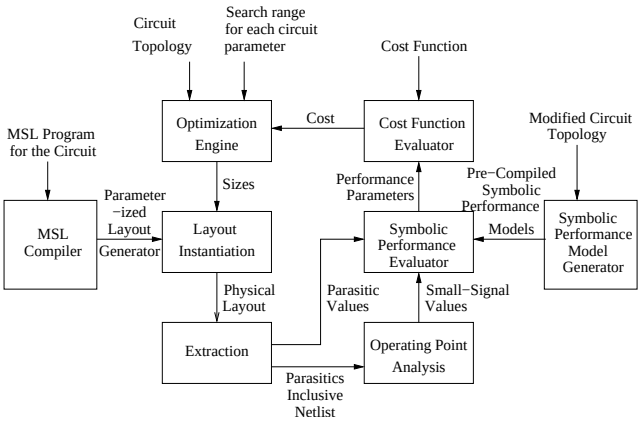

Fig 1. Proposed Approach

executable layout generator which can be quickly elaborated into a concrete layout, when concrete size information is available from the optimization engine. A library of parameterized module generators, including fingered transistors, differential pairs, current mirrors, resistors, capacitors,inductors etc, has been developed in MSL.

2.2. Symbolic Performance Modeling

Symbolic performance models (SPMs) are symbolic equations in terms of circuit parameters [9]. They represent the characteristics of an analog circuit. SPMs are built using symbolic transfer functions which are obtained by symbolic circuit analysis. In circuit synthesis, SPMs are used for repetitive performance estimation during the optimization iterations. Unlike numerical simulators, analysis is not done in every iteration and only evaluation of the SPMs is needed. This results in a significant speedup of the performance estimation time.

The framework for the generation of SPMs is presented in Figure 2. The first step is to generate combinations of nodes that appear in a performance characteristic formula. All the active devices in a circuit are expanded to their small-signal models. The symbolic analysis engine uses the node information to generate the required transfer functions as Element Coefficient Diagrams (ECDs). The symbolic model builder uses the node information to generate the formulae for the desired performance characteristics. The combination of the symbolic formulae and transfer functions are called SPMs. In our methodology, the SPMs are generated using a modified parasitic-inclusive circuit topology. Techniques to include the relevant set of parasitic elements and various layout effects in the SPMs are described in Section 3.

The core of this SPM generation process is *symbolic analysis*. Symbolic analysis is a formal technique used to obtain network transfer functions in terms of symbolic circuit parameters and independent variables like frequency [10]. In circuit synthesis it is essential that the SPMs are not approximate, require minimum space, are fast to evaluate and are stored in the *s-polynomial* format [7]. In this paper we use ECDs to represent a symbolic determinant. The process of ECD-based symbolic analysis is described in [11]. The ECDs are converted to

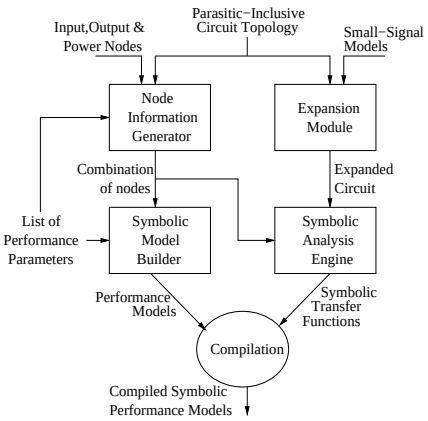

Fig 2. Symbolic Performance Model Generation

C++ code and then compiled. The use of pre-compiled ECDs reduces the SPM evaluation time considerably. An average speedup by a factor of 32, with respect to evaluation of ECDs stored in memory, was observed for for the five benchmark circuits discussed in this paper.

3. Inclusion of Layout Effects in Symbolic performance Models

As component sizes vary during synthesis, the layout geometry varies between iterations. This variation may generate varying sets of parasitic elements (resistances and capacitances) in each iteration. The variation of width of a transistor module also causes the number of fingers to change between iterations. For example, Figure 3 shows two instances of a transistor module with and without fingers. In this section we talk about techniques to model these variations in the original circuit topology, which in turn generates parasitic-aware SPMs.

3.1. Inclusion of Layout Parasitic Elements

Let $C(R)$ be the set of all parasitic capacitances (resistances) ever appearing in an extracted circuit. Some elements of $C(R)$ might be missing (set to zero) in some instances of the extracted circuits. However, an SPM including all potential parasitic capacitances (C) and resistances (R), must be pregenerated. Techniques to do so are described next.

3.1.1 Inclusion of Parasitic Capacitances

We describe three techniques for determining and including parasitic capacitances, C, in the SPMs.

1. *Complete Set Technique*

This method assumes that there is a potential symbolic capacitance between every two nodes of the circuit. Therefore,

$$C = \{C_{ab} = C_{ba} \,|\, a, b \in N, a \neq b\}$$

where N = set of all nodes. Hence $|C| = \mathrm{C}_2^{|N|} = \frac{|N|(|N|-1)}{2}$ The advantage of using this technique is that no parasitic

element will be missed out and hence the results obtained by evaluation of the models will be accurate. However, their is an explosion in the number of circuit elements due to the inclusion of symbolic parasitic elements some of which may never be extracted from any layout. The result is an increase in the time required to generate and compile the ECDs.

2. *Layout Sampling Technique*

In layout sampling technique, we generate a number of sample layouts of the entire circuit, examine which area and coupling capacitances are extracted from these layouts and symbolically include only those capacitances in the SPM generation process. The advantage of using this method is that only relevant parasitics, i.e., those which have appeared due to extraction of layouts, are taken into account. This method may not generate all possible parasitic elements and this may cause inaccuracy if a new physical capacitor is extracted. There is an additional overhead of generating the layout samples. The sampling methodology used in our experiments is random sampling.

3. *Analysis-based Technique*

This method uses information about the layout-template and routing technique and determines the symbolic parasitic capacitances. During layout extraction, the two significant groups of capacitances extracted are *area* and *internodal coupling* capacitances [12]. The internodal coupling capacitances include overlap, sidewall overlap and sidewall capacitances. In any layout the two main parts are *modules* and *interconnects*. Each module and interconnect has both area and coupling capacitances. Hence, the four categories of capacitances that are generated after extraction are: C_{MA}, the set of area capacitances of all modules; C_{MC}, the set of coupling capacitances of all modules; C_{IA}, the set of interconnect area capacitances; C_{IC}, the set of interconnect coupling capacitances. The complete set of parasitic capacitances is:

$$C = C_{MA} \cup C_{MC} \cup C_{IA} \cup C_{IC}$$

(a) *Module Capacitances Analysis*

This involves determining C_{MA} and C_{MC}. In an MSL system a module has fixed templates. A fixed template always generates the same set of capacitances for any size. The set of these potential capacitances can be determined either by applying the knowledge of extraction rules or producing the layout of any one instance of the module. Figure 3 shows two templates of a transistor module, one with fingers and the other without. For the case of non-fingered transistor module (module x) the set of capacitances obtained are:

$$C_{MA_x} = \{C_{n1\_GND}, C_{n2\_GND}, C_{n3\_GND}, C_{n4\_GND}\}$$

$$C_{MC_x} = \{C_{n1\_n2}, C_{n1\_n3}, C_{n1\_n4}\}$$

For the fingered module the set of all possible capacitances is different from previous example because a few extra capacitances have been introduced due to internodal interconnect coupling. All possible symbolic capacitances in this case are:

$$C_{MA_y} = \{C_{n1\_GND}, C_{n2\_GND}, C_{n3\_GND}, C_{n4\_GND}\}$$

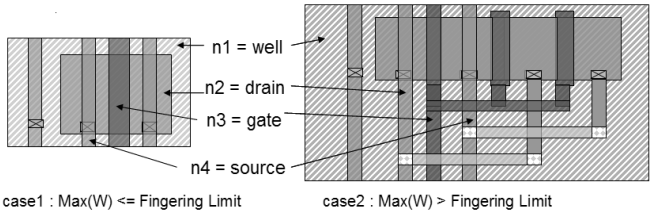

Fig 3. Transistor Modules

$$C_{MC_y} = \{C_{n1\_n2}, C_{n1\_n3}, C_{n1\_n4}, C_{n2\_n3}, C_{n2\_n4}, C_{n3\_n4}\}$$

Since the template information for all modules is available, the set of all possible capacitances can be determined for the entire layout. Finally we have

$$C_{MA} = \bigcup_{i=1}^{m} C_{MA_i} \quad \& \quad C_{MC} = \bigcup_{i=1}^{m} C_{MC_i}$$

where, m is the number of modules. Eventually $|C_{MA}| = |N| - 1$, since each node has an area capacitance to the substrate.

(b) *Interconnect Capacitances Analysis*

The MSL system uses a channel routing style to route the nets and the technique described below is specific only to that style. Similar to the case of modules, the set of area capacitances and internodal coupling capacitances for the interconnects have to be found. The set of area capacitances for interconnects is the same as that of the modules $C_{IA} = C_{MA}$, since all interconnect nodes are present inside the modules too. Therefore, one symbolic capacitance at a node can model multiple physical capacitances at that node, one due to the node layer and others due to the interconnect. At the SPM evaluation time all the parallel physical capacitor values are added up to yield one value for the symbolic capacitance.

For the internodal coupling capacitances of interconnects, we focus only on the sidewall coupling capacitances because they are the only significant ones. The coupling capacitances are obtained as described below. Let, h be the number of channels and C_{IC_i} the set of internodal coupling capacitances in channel i. Then, $C_{IC} = \bigcup_{i=1}^{h} C_{IC_i}$. The capacitances in each channel are very difficult to estimate, since based on the modules sizes the routing varies and hence also the coupling between interconnects. Therefore for each channel we include a complete set of all-possible capacitances. For each channel this set is known, as we are familiar with the layout template. Let, n_i is the set of nodes in channel i. Then

$$C_{IC_i} = \{C_{xy} = C_{yx} \,|\, x, y \in n_i, x \neq y\} \;\&\; |C_{IC_i}| = \frac{|n_i|\,(|n_i| - 1)}{2}$$

The advantages of using this method are that all the relevant capacitances are obtained without generating sample layouts. Some unnecessary capacitances are generated because of the exhaustive technique used for interconnect coupling capacitances. The limitation of this method is that it is valid for a fixed layout template methodology.

Table 1. Comparison of Parasitic Capacitances Inclusion Techniques

Circuit Name	# Devices	# Nodes	# Parasitic Caps			ECD Depth	# ECD-Vertices			# ECD-Edges		
			CS	LS	AB		CS	LS	AB	CS	LS	AB
SEO1	9	10	55	32	37	10	705	503	580	8,694	3,972	5,565
TSO1	5	11	55	31	31	11	1,236	668	668	14,352	4,234	4,234
TSO2	9	11	55	30	32	11	1,550	1,024	1,288	25,953	8,479	11,996
SEO2	16	17	136	56	65	17	264,254	1,661	23,450	5,109,736	13,522	222,789
LPF	22	22	231	48	61	22	3,734,349	11,583	80,326	24,294,879	71,563	613,667

Table 2. Time Comparisons for Parasitic Capacitances Inclusion Techniques

Circuit Name	ECD Generation			ECD Compilation			ECD Evaluation		
	CS	LS	AB	CS	LS	AB	CS	LS	AB
SEO1	9.4s	8.8s	9.3s	17.0s	10.0s	12.8s	1.6ms	1.0ms	1.2ms
TSO1	9.6s	8.9s	8.9s	22.2s	10.3s	10.3s	2.5ms	1.4ms	1.4ms
TSO2	10.3s	9.2s	9.3s	37.9s	15.2s	19.3s	3.1ms	2.1ms	1.5ms
SEO2	252.3s	10.7s	19.1s	C/F	30.6s	53m18.1s	C/F	3.2ms	19.3ms
LPF	8323.3s	13.8s	43.1s	C/F	212.8s	C/F	C/F	17.7ms	C/F

Table 1 and Table 2 compare the complete-set (CS), layout sampling (LS) and analysis-based (AB) techniques of layout parasitic capacitances. The comparison is done on five benchmarks. Single-Ended Op-amp1(SEO1) is the device model of CMOS Operational Amplifier in [13]. Twostage Op-amp2(TSO2) is borrowed from page 308 of [14]. Low Pass Filter (LPF), has been borrowed from page 410 of [15]. LPF is a fourth-order Butterworth filter of Sallen-Key implementation. SEO2 is also a single-ended op-amp and TSO1 is yet another twostage op-amp. SEO1 is used to implement the LPF.

Table 1 gives an account of the number of capacitances added to the original topology for each inclusion technique. It also demonstrates the effect of number of extra capacitances on the size of the ECDs. Table 2 shows the generation, compilation and evaluation times of ECDs (and hence the SPMs). For SEO2's CS technique and LPF's CS and AB techniques the compiler crashed (C/F = Compiler Failure) while compiling the ECDs due to the large size of circuits. This implies that for large circuits the LS technique is the most suitable technique. As the size of circuits increase further, hierarchical techniques of symbolic analysis will have to be used for generation of SPMs.

3.1.2 Inclusion of Parasitic Resistances

In this section we describe a technique to include parasitic resistances in the SPMs. Resistance extraction is done only for the interconnects of the layout. The analysis-based parasitic capacitance inclusion technique is not valid for interconnects when resistance extraction is done. Hence a two stage extraction strategy is employed. First only the modules are extracted to obtain the symbolic parasitic capacitances. Then the interconnects are extracted to obtain both symbolic parasitic resistances and capacitances. The parasitic resistances can be modeled as described below. The interconnect parasitic capacitances are modeled as a capacitance to ground for each node extracted during symbolic resistance extraction. For multi-terminal nets the number of resistances extracted in different iterations may not be the same. This is because the pin positions change and this causes the nets geometry to change. Figure 4 shows a multi-terminal net with one terminal on the top row and three terminals on the bottom row. Based on the relative *x*-coordinate of node a, *a(x)* with respect to *c(x)*, the number and position of resistances extracted vary. To model these variations a complete model, which is a superset of all possible combinations, is used to generate the SPMs. The actual evaluation of the SPMs is done by determining the relative coordinates of the critical nodes (*a* and *c*, in the above example) and setting the value of some resistances and conductances to zero. Extending the above example to the general case where a multi-terminal net has *m* terminals in the top row and *n* terminals in the bottom row, Figure 5 shows the complete resistance network used in the circuit topology for the generation of SPMs. In this network there is a set of resistances between each intermediate node of the top terminals and intermediate node of the bottom terminals called R_{INT}.

$$R_{INT} = \{R_{ab} = R_{ba} \,|\, a \in T_N \; b \in B_N\}$$

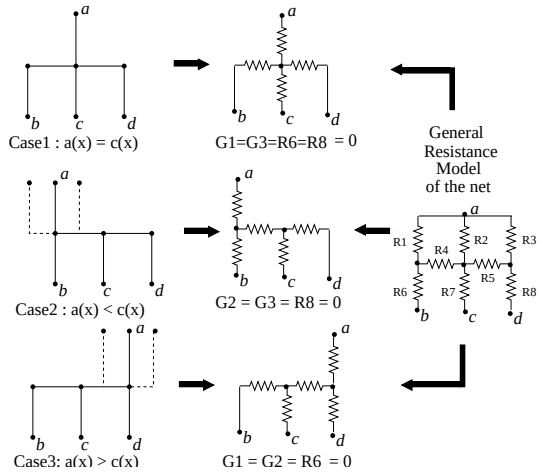

Fig 4. An Example of Inclusion of Parasitic Resistances

Table 3. Layout-Inclusive Circuit Synthesis Results

Circuit	Attribute	Constraints	Flow1	Flow2	Flow2 Verified	% Error
SEO1	*DC Gain*	≥50 dB	50.133 dB	54.185 dB	54.182 dB	+0.006%
	F_{-3dB}	≥1.0e+05	1.761e+05	1.318e+05	1.308e+05	+0.759%
	UGF	≥1.0e+07	5.601e+07	6.348e+07	6.459e+07	-1.749%
	PM	≥ 60°	68.401°	64.273°	63.888°	+0.599%
TSO1	*DC Gain*	≥40 dB	40.568 dB	43.921 dB	43.42 dB	+1.141%
	F_{-3dB}	≥5.0e+06	5.034e+06	5.033e+06	5.037e+06	-0.079%
	UGF	≥5.0e+07	5.430e+08	6.908e+08	6.659e+08	+3.605%
	PM	≥ 50°	61.389°	54.961°	54.042°	+1.672%
TSO2	*DC Gain*	≥23 dB	23.918 dB	25.721 dB	25.948 dB	-0.883%
	F_{-3dB}	≥5.0e+06	8.998e+06	5.015e+06	5.238e+06	-4.447%
	UGF	≥5.0e+07	1.257e+08	1.049e+08	1.022e+08	+2.574%
	PM	≥ 50°	50.434°	74.956°	77.899°	-3.926%
SEO2	*DC Gain*	≥35 dB	42.665 dB	40.037 dB	39.774 dB	+0.657%
	F_{-3dB}	≥1.0e+06	1.021e+06	1.142e+06	1.181e+06	-3.415%
	UGF	≥1.0e+07	1.196e+08	8.451e+07	8.722e+07	-3.207%
	PM	≥ 60°	64.39°	75.308°	76.73°	-1.888%
LPF	*DC Gain*	≥10 dB	10.119 dB	10.201 dB	10.201 dB	0.00%
	F_{-3dB}	950-1050	1042.85	1025.04	1024.8	+0.023%

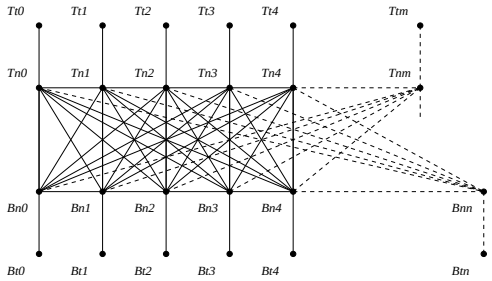

Fig 5. General Inclusion of Parasitic Resistances

where, T_N is the set of all intermediate nodes corresponding to the top row and B_N is the set of all intermediate nodes corresponding to the bottom row

Another set of resistances R_{ADJ} exist between each pair of adjacent nodes of T_N and each pair of adjacent nodes of B_N. There is also a set of resistances between each intermediate node of T_N and B_N, and the corresponding terminal nodes in T_T and B_T. This set of resistances is referred to as R_{TER}. Thus,

$$R = R_{INT} \cup R_{ADJ} \cup R_{TER}$$

During evaluation based on the pin position of top terminals with respect to the bottom terminals, most of the resistances are eliminated by setting their conductances to zero.

3.2. General Inclusion of Fingered Transistor Effects

A transistor module is expanded to a small signal model before symbolic analysis. Depending on the width of the transistor it may or may not be fingered during an iteration. When a transistor is fingered, it appears as more than one transistor in the extracted file. This variation in the number of fingers needs to be captured in the SPMs.

A transistor module is modeled by only one lumped small-signal model, even if is fingered. The parameters of the lumped small-signal is just a summation of small-signal values of each finger. Hence, $C_{gs} = \sum_{i=0}^{f} C_{gs_i}$ and other lumped small-signal parameters can be expressed similarly. Here f is the number of fingers. The small-signal values of each finger are obtained by doing a numerical simulation using SPICE. To eliminate the need to dig up the small signal parameter value of each finger from the simulation results, only the values of the first, second and the last finger(in case the last finger is a of different size than others) are obtained. If the number of fingers is even and all fingers are of the same size then (other cases can be expressed similarly), then the lumped values are: $C_{gb} = f \times C_{gb_1}, G_m = f \times G_{m_1}, G_{mb} = f \times G_{mb_1}, G_{ds} = f \times G_{ds_1}$ and, $C_{gs} = f/2 \times C_{gs_1} + f/2 \times C_{gd_2}$, $C_{gd} = f/2 \times C_{gd_1} + f/2 \times C_{gs_2}$

4. Experimental Results

Table 3 presents the results for two approaches of circuit synthesis.The first approach (Flow1) uses the MSL environment and NG-Spice for layout-inclusive synthesis. The second approach (Flow2) uses MSL with SPMs. The layout sampling technique is used for the inclusion of parasitic capacitances. The layout editor used in both approaches is Magic 7.1 and the numerical simulator is NGSpice. Parasitic resistances have not been modeled in these experiments. The obtained performance estimates for Flow2 are verified using NG-Spice. All the results are within 5% error. This error can be attributed in part to the LS parasitic inclusion technique as it fails to include all parasitic capacitances and in part to the bisection method of solving for roots of the transfer function, to obtain the values of unity gain and -3dB frequencies.

Table 4 presents the time results for both methodologies. The performance convergence in both cases is fast due to the use of a language-based layout generator. Flow2 is 15%-30% faster than Flow1, per iteration. The Numeric analysis time of Flow2 (for operating point analysis) is 65%-75% faster than Flow1. Model evaluation time of Flow2 (SPM evaluation time) is 13-88 times faster than model evaluation (the process of obtaining desired performance attributes from the

Table 4. Time Results of Layout-Inclusive Circuit Synthesis

Circuit Name	Synthesis Approach & Speedup	Total System Time for Synthesis	# Iterations	Time per Iteration				
				Layout Generation & Extraction	Performance Estimation			Total Iteration Time
					Numerical Analysis	Model Evaluation	Total	
SEO1	Flow1	520.2s	286	1.039s	0.441s	0.158s	0.603s	1.642s
	Flow2	125.7s	96	1.042s	0.254s	0.0018s	0.263s	1.309s
	Speedup	N/A	N/A	-0.3%	73.6%	8683%	129.3%	25.4%
TSO1	Flow1	893.3s	432	1.296s	0.447s	0.171s	0.624s	1.919s
	Flow2	361.1s	230	1.292s	0.264s	0.0026s	0.273s	1.570s
	Speedup	N/A	N/A	0.03%	69.9%	6547%	128.6%	22.2%
TSO2	Flow1	1865.2s	902	1.311s	0.425s	0.187s	0.621s	1.932s
	Flow2	932.1s	585	1.311s	0.248s	0.0023s	0.257s	1.593s
	Speedup	N/A	N/A	0.00%	71.4%	8042%	141.6%	21.3%
SEO2	Flow1	1476.6s	446	1.703s	0.461s	0.171s	0.643s	2.346s
	Flow2	1010.0s	369	1.704s	0.273s	0.0041s	0.285s	1.989s
	Speedup	N/A	N/A	- 0.06%	68.9%	4081%	125.6%	18.0%
LPF	Flow1	524.3s	253	1.095s	0.417s	0.248s	0.671s	1.766s
	Flow2	692.2s	503	1.093s	0.241s	0.019s	0.271s	1.374s
	Speedup	N/A	N/A	0.1%	73.0%	1210%	147.6%	28.5%

analysis results) time for Flow1. Overall, SPM-based performance estimation time is 1.25-1.5 times faster than NG-Spice-based method. The total synthesis time for both approaches is not compared because the number of iterations to converge to a solution is different in both cases (N/A is Not Applicable). All experiments have been conducted on SunBlade 1000 with Solaris(SunOS), 2048MB RAM and 2-750MHz Processors.

5. Conclusions and Future Work

The paper has proposed a new methodology for layout-inclusive circuit synthesis. The use of SPMs speeds up the simulation process considerably and the overall synthesis speed is faster by 15%-30%. The evaluation of pre-compiled SPMs is significantly faster than those in memory. Techniques have been developed to accurately and efficiently model the layout effects in the SPMs. There are two main areas where the proposed methodology can be improved. First, the size of circuits this method can handle are not very large. This limitation can be remedied by using hierarchical symbolic analysis techniques. Second, the SPMs in this paper are limited to modeling AC behavior of benchmark circuits. With recent progress in symbolic analysis of circuits exhibiting hard non-linearity, this technique can be extended to non-linear circuits and models and ultimately to the automatic synthesis of RF circuits. As a part of the on-going work we are developing techniques to model circuit and parasitic inductors in the SPMs. Further research has to be done for the symbolic modeling of non-transfer function related performance metrics. We are also exploring techniques to include layout parasitics for different placement and routing styles. This work is a part of the Synthesis for High Performance Analog and RF Circuits (SHARC) project (http://www.ececs.uc.edu/˜ddel).

References

[1] G. Gielen and R. Rutenbar. Computer-aided design of analog and mixed-signal integrated circuits. *Proc. IEEE*, 88(12):1825–1854, Dec. 2000.

[2] M. Dessouky and et al. "Layout-oriented synthesis of high performance analog circuits". In *Proc. of DATE Conf.*, pages 53–57, 2000.

[3] P. Vancorenland and et al. "A layout-aware synthesis methodology for rf circuits". In *Proc. of ICCAD*, pages 358–362, 2001.

[4] W. Daems and et al. Simulation based automatic generation of signomial and posynomial performance models for analog integrated circuit sizing. In *Proc. of ICCAD*, pages 70–74, 2001.

[5] P. Mandal and V. Vishvanathan. CMOS op-amp sizing using a geometric programming formulation. *IEEE Trans. on CAD*, 20:22–38, Jan. 2001.

[6] G. Wolfe and R. Vemuri. Extraction and use of neural network models in automated synthesis of operational amplifiers. *IEEE Trans. on CAD*, 22(2):198–212, Feb. 2003.

[7] C.-J. R. Shi and X.-D. Tan. Compact representation and efficient generation of s-expanded symbolic network functions for computer-aided analog circuit design. *IEEE Trans. on CAD of Integrated Circuits and System*, 20(7):813–827, July 2001.

[8] H. Sampath and R. Vemuri. Msl: A high-level language for parameterized analog and mixed signal layout generators. In *12th Intl. IFIP VLSI Conf.*, 2003.

[9] G. Gielen and et al. Analog circuit design optimization based on symbolic simulation and simulated annealing. *IEEE Jour. of Solid-State Circuits*, 25:707–713, June 1990.

[10] G. Gielen and W. Sansen. *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Kluwer Academic Publishers, 1991.

[11] W. Verhaegen and G. Gielen. Efficient DDD-based symbolic analysis of large linear analog circuits. In *Proc. of DAC*, pages 139–144, 2001.

[12] W. S. Scott and J. K. Ousterhout. Magic's circuit extractor. In *Proc. of DAC*, pages 286–292, 1985.

[13] B. Kaminska and et al. "Analog and mixed-signal benchmark circuits first release". In *Proc. of IEEE Intl. Test Conf.*, pages 183–190, 1997.

[14] B. Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw Hill, 2001.

[15] A. Nunez-Aldana and R. Vemuri. An analog performance estimator for improving the effectiveness of cmos analog systems circuit synthesis. In *Proc. of DATE Conf.*, pages 9–12, 1999.