Fast and Accurate Parasitic Capacitance Models for Layout-Aware Synthesis of Analog Circuits*

Anuradha Agarwal, Hemanth Sampath, Veena Yelamanchili and Ranga Vemuri
Department of ECECS, University of Cincinnati
Cincinnati, OH 45221 USA

{agarwala,hsampath,vyelaman,ranga}@ececs.uc.edu

ABSTRACT

Considering layout effects early in the analog design process is becoming increasingly important. We propose techniques for estimating parasitic capacitances based on look-up tables and multi-variate linear interpolation. These models enable fast and accurate estimation of parasitic capacitances and are very suitable for use in a synthesis flow. A layout aware methodology for synthesis of analog CMOS circuits using these parasitic models is presented. Results indicate that the proposed synthesis system is fast as compared to a layout-inclusive synthesis approach.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: Design Aids

General Terms

Algorithms

Keywords

Analog Synthesis, Layout Aware, Parasitic Estimation

1. INTRODUCTION

The behavior of analog circuits is extremely sensitive to layout-induced parasitics. Parasitics not only influence the circuit performance but often render it non-functional. Hence, it is essential to consider the effect of parasitics early in the design process. Traditionally, the circuit synthesis step is followed by layout synthesis and each step is carried out independent of the other. This is followed by a verification step to check whether the desired performance goals have been achieved after layout generation and extraction. These steps are carried out iteratively till the desired performance goals are met. This approach is extremely time-consuming and no structured feedback from previous runs can be readily used to re-design the circuit if the layout fails to meet performance goals.

One way of performing layout aware synthesis is to perform layout synthesis and extraction inside the circuit synthesis loop as shown in Fig. 1a [1, 2, 3]. Since the layout generation and extraction process is moved inside the optimization loop, fast procedural layout generators (PLGs) are generally used.

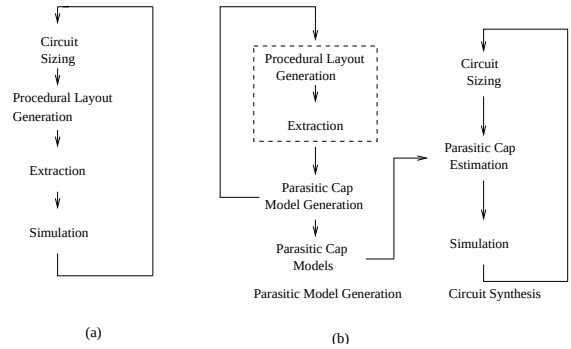

Figure 1: (a) Layout-in-the-loop Approach (b) Proposed Approach

Onodera et al.[1] introduced layout effects in the synthesis flow with the use of a PLG. Extraction functions were specified as procedures in the layout environment. In the circuit optimization loop, a layout is generated and the parasitic elements are extracted using the pre-defined extraction procedures. The extraction functions used are relatively simple and calculate the wire and diffusion area capacitances. Dessouky et al.[2] have extended the approach in [1] by further enhancing the layout generation environment and considering a more complex parasitic extraction model. However, the parasitic model included only area capacitances of diffusion, well and interconnect. Overlap capacitances between intra-module wires was also calculated. Lateral capacitances inside the module and the coupling capacitances (lateral and overlap) of the inter-module interconnect was neglected. Vancorenland et al.[3] used a PLG for layout generation in the optimization loop. More accurate extraction was achieved in this case. However, the accuracy was achieved at the cost of actual layout generation and detailed extraction. Choi et al.[4] have modeled the parasitics associated with inductors during RF circuit synthesis. Other parasitics have however not been accounted for.

This paper presents an alternative way of performing layout-aware synthesis based on accurate estimation of layout parasitics *without* performing layout generation as shown in Fig. 1b. Our method ensures the use of PLGs which are essential to the flow in Fig. 1. However, instead of using them in the synthesis loop, we build parasitic macro-models by analyzing several instances of the layouts produced by PLGs. These macro-models are used in the synthesis loop to yield fast and accurate parasitic estimates. We

*This work is sponsored in part by the the DARPA/MTO NEOCAD program under contract number F33615-01-C-1977 monitored by the Air Force Research Laboratory.

DAC 20004, June 7–11, 2004, San Diego, California, USA.

propose a synthesis flow incorporating the parasitic macro-models and show that this flow is capable of achieving designs with layout closure. In this work, we have modeled parasitic capacitances as in [1, 2].

The rest of the paper is organized as follows. In Section 2, parasitic capacitances are classified into several categories and a model is developed for each. Section 3 discusses the impact of various parasitic capacitances on performance of analog circuits. In Section 4, a layout-aware synthesis methodology is proposed where the use of the developed parasitic models is demonstrated. Experimental results have been presented in section 5. Finally, Section 6 concludes the paper.

2. MODELING OF PARASITICS

In order to classify circuit parasitics, we introduce the concept of modules. We define a module to be a device or a group of devices that are laid out as one entity. Transistors, inductors, differential pairs, current mirrors, etc. are examples of modules. We also classify layout synthesis for analog circuits into four broad steps: circuit partitioning into modules, module generation, module placement and inter-module routing. We have classified the parasitic capacitances into two categories for the purpose of modeling:

- Intra-module parasitics, which include all parasitics internal to a module. We further classify these parasitics (based on the way they are extracted) into diffusion and non-diffusion parasitics.
- Inter-module interconnect parasitics, which are determined by the placement and routing steps and represent the parasitics associated with the inter-module interconnect.

2.1 Intra-module Parasitic Modeling

A procedure layout generation environment called MSL (Module Specification Language)[5] is used to specify layout generators. MSL is an attribute based high level language which allows hierarchical composition of layout primitives and permits their relative placement. An MSL program specifies a parameterized layout generator. When the parameter values(typically device sizes) are supplied, a concrete layout is obtained. Off-the-shelf extractors can be used to extract a circuit including parasitic elements from this layout. Fig. 2 shows two instances of a fingered transistor generated using MSL and the corresponding extracted circuits are shown in Fig. 3. The classification of the parasitic capacitances in Fig. 2 is shown in Table 1.

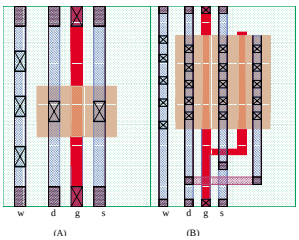

Figure 2: Fingered Transistor Layout (a) 1μ (b) 10μ

2.2 Diffusion Capacitance Modeling

Extractors usually extract the area and perimeter of the drain and source of each transistor. Simulators use these values (named AD, PD, AS, PS in spice transistor models) to calculate the diffusion parasitics internally. Generally in case of fingered devices, the diffusion parasitics of the entire device is lumped and attached

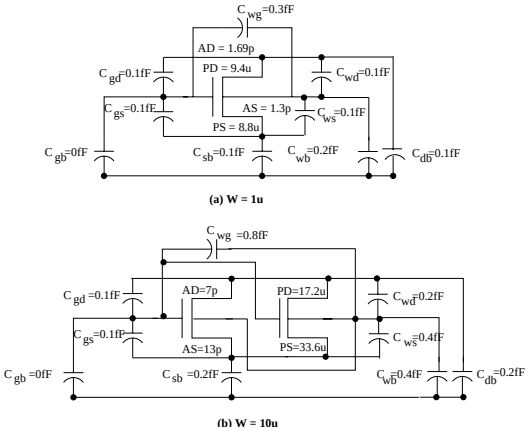

Figure 3: Extracted Circuits for Fingered Transistor in TSMC 0.18μ

Table 1: Classification of Parasitic Caps for Fingered Tran.

Capacitance	Classification
C_{db}	Interconnect AreaCap
C_{gb}	Gate(Poly) AreaCap
C_{sb}	Interconnect AreaCap
C_{wb}	Well AreaCap
C_{gs}	Interconnect Coupling Cap
C_{gd}	Interconnect Coupling Cap
C_{wd}	Well-Interconnect Overlap Cap
C_{ws}	Well-Interconnect Overlap Cap
C_{wg}	Well-Gate Overlap Cap

to the first finger in the extracted circuit. The diffusion parasitics for several layout instances is gathered using a uniform grid based sampling technique and stored in a Module Characterization Table (MCT). The MCT is a look-up table with input and output columns. The inputs are usually the transistor sizes (or other size parameters depending on the module). The output columns comprise of the extracted parasitics. Data in MCTs can be used to construct a variety of macro-models such as neural networks[6], posynomials[7] and polysplines[6], etc. In this paper, we adopted table look-up and linear interpolation. We found this method to be fast and accurate in practice, as will be demonstrated by the experimental results.

For each module, one model was developed for transistors with odd number of fingers and another for transistors with even number of fingers. During table lookup, depending on whether the query variable was odd or even, the corresponding model was chosen. The diffusion parasitics depend not only on the transistor width and the constant diffusion width but also on the number of fingers (N_f). The number of fingers determine the number of source (N_s) and drain regions (N_d), for instance in Fig. 2a, the transistor has a single finger and $N_d = N_s = 1$. The transistor in Fig. 2b has two folds and in this case, $N_d = 2$ and $N_s = 1$. Equations 1 and 2 denote the relationship between N_d, N_s and N_f when N_f is even and odd respectively. This justifies the need for two separate models.

$$N_s = \frac{N_f}{2}$$
$$N_d = \frac{N_f + 2}{2} \quad (1)$$

$$N_s = N_d = \frac{N_f + 1}{2} \quad (2)$$

In order to validate our models, we compared the values of the diffusion parasitics estimated by the proposed technique with that obtained using an off-the-shelf extractor for five different modules. In each case, the input variable was the width of the transistor. There were 500 samples in the MCT and 200 validation points were generated randomly. The time required for generating the MCT for all intra-module parasitics is presented in the second column of Table 2. Some parts of Tables 2 and 3 have been published in [8] and have been included here for completeness.

Table 2: Diff Cap Deviations for Modules

Circuit	MCT Gen Time(s)	Mean Dev(%)	Max Dev(%)	Std Dev(%)
Fingered Tran(n)	1150	0.05	1.4	0.19
Fingered Tran(p)	1150	0.09	1.33	0.24
Current Mirror	1250	0.09	1.33	0.24
Diff Pair	1250	0.21	2.45	0.38
Diff Pair Src Gate Connected	1300	0.24	1.63	0.38

2.3 Non-Diffusion Capacitance Modeling

Extractors extract explicit capacitive parasitics associated with non-diffusion layers. Parasitics comprising this category are the well, gate and interconnect parasitics (area as well as coupling). Table 1 showed these capacitances for a fingered transistor module. For each module in the library, a MCT is generated from several layout instances of the module. During synthesis, these MCTs are used for predicting the capacitance values. We generated two separate models for each device, one with odd number of fingers and another with even number of fingers for the same reason as described in section 2.2.

Table 3 shows experimental data illustrating the accuracy of the MCT tables for non-diffusion capacitances. The table size was 500 and the number of validation points were 200.

2.4 Inter-module Capacitance Modeling:

The inter-module wire parasitics (area as well as coupling) are dependent on the placement and routing of the modules in the circuit. In MSL, the relative placement is defined with the help of positional attributes in the language. Once the modules have been placed, the routing between them can be accomplished with the help of routing boxes in MSL. A routing box constitutes of a horizontal wire and several vertical wires as shown in Fig. 4. A1, A2 and A3 are three terminals that need to be connected. For various layout instances, the relative positions of A1 with respect to A2 and A3 will be different. The routing box module determines the leftmost and rightmost terminal positions once the absolute location of all terminals has been determined. It then draws a horizontal wire spanning the leftmost and rightmost terminals. It next drops a vertical connection from each terminal to the horizontal wire for obtaining connectivity between them. Each net in the layout is defined using a routing box. The user however has the burden of ensuring the correctness of the layout for various sizes.

Table 3: Non-Diff Cap Deviations for Modules

Circuit	MCT Gen Time(s)	Mean Dev(%)	Max Dev(%)	Std Dev(%)
Fingered Tran(n)	1150	0.3	2.9	0.52
Fingered Tran(p)	1150	0.19	2.66	0.3
Current Mirror	1250	0.25	2.62	0.24
Diff Pair	1250	0.21	3.15	0.23
Diff Pair Source Gate Connected	1300	0.32	4.12	0.75

Table 4: Inter-module Interconnect Model Data

Circuit	No. of Variables	No. of Data Points	Table Gen Time(s)
RC	4	81	201
TSO	3	27	47
SEO	7	2187	5247
BPF	3	27	57

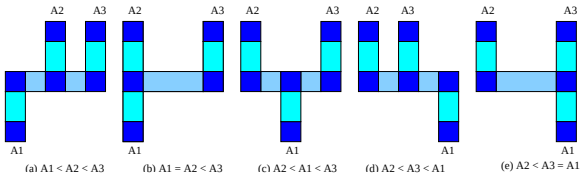

Figure 4: Instances of a Routing Box

Modeling the inter-module interconnect involves two steps: *MCT Data Generation* and *Multi-Variate Interpolation.*

For MCT data generation each variable (width) in the circuit is sampled on a uniform grid. Hence, if we sample m points for each of n variables, the total number of data points would be m^n. As the total number of data points and hence the time required for data generation grows exponentially with the number of variables, this technique would become extremely time consuming for more than 3 samples for 7-8 variables. Since most opamp-sized circuits can be reduced to 5-6 variables due to matching among devices, this methodology would work well for such circuits. Table 4 depicts the data generation details for the four benchmarks. MCT table lookup procedure with multi-variate interpolation comprises of the following steps.

Let q be the query point, n the number of variables in the query point.

1. If the query point, q is defined in the MCT, return the corresponding parasitics and quit.
2. In each dimension(D) of q, determine the 2 nearest neighbors. Combine the neighbors in each D to obtain 2^n neighbors of q. Store the parasitic data corresponding to them in table(ParDataTable).
3. Perform Linear Interpolation between every two points of ParDataTable which differ from each other in the value of only one variable(say ith). Replace these two entries in ParDataTable with a single entry with the interpolated values of the parasitics and the variable set to q[i].
4. Continue step 3 until interpolation has been performed with respect to all variables and the ParDataTable has only one entry. Return the parasitic data corresponding to this entry and quit.

To validate the interconnect capacitance estimation method, we considered four analog circuits: Single Ended Opamp (SEO), Two-Stage Opamp (TSO), Regenerative Comparator (RC) and Band-pass Filter (BPF) shown in Fig. 5, developed parameterized layout generators for each in MSL and generated MCTs for all inter-module interconnect parasitic capacitances. We then generated a large number of random validation points (sizes) for each circuit, used interpolation to estimate the capacitances and compared them with actual values obtained by concrete layout generation and extraction. Fig. 6 shows the validation results for the four circuits combined. Total number of validation points was 500.

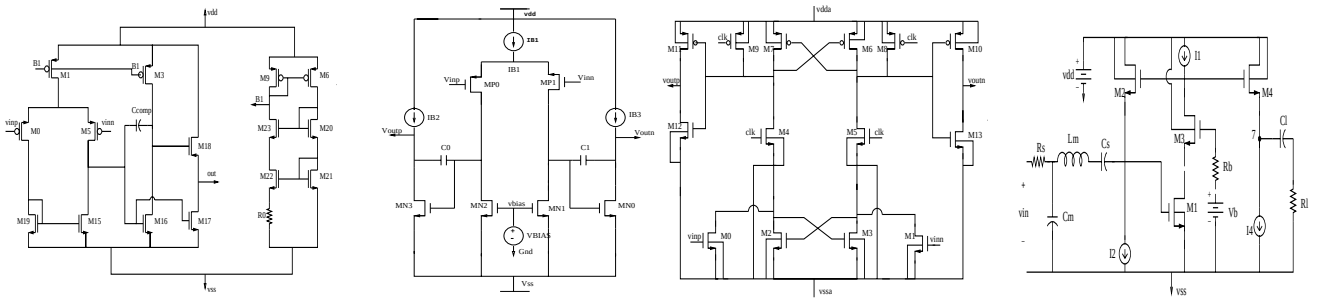

Figure 5: (a)Single Ended Opamp (b) TwoStage Opamp (b) Regenerative Comparator (c) Bandpass Filter

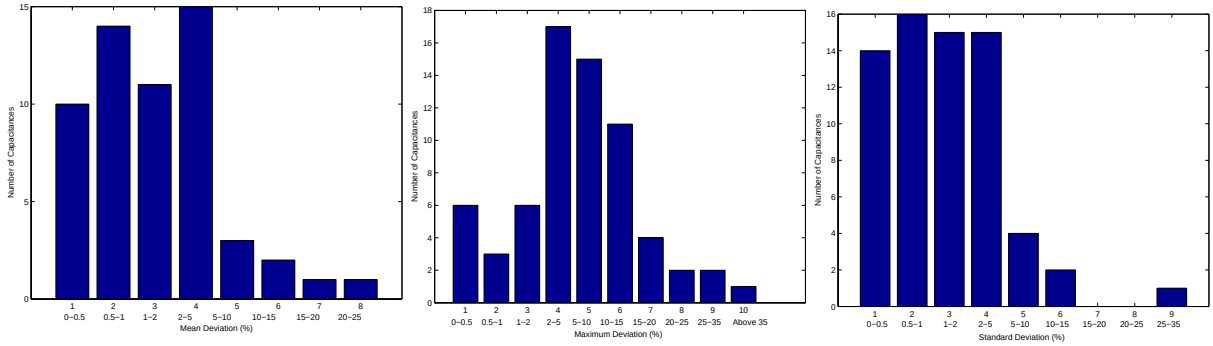

Figure 6: Interconnect Capacitance Deviations For All Benchmarks in %

Table 5 shows the validation details for one of the circuits, the regenerative comparator. This table provides the mean, maximum and standard deviations for each of the inter-module interconnect capacitances. The first field in the table denotes the nodes between which the parasitic capacitance appears. The next three fields indicate the mean, maximum and standard deviation in percentage. We observed that the area caps generally show less deviation as compared to the inter-nodal coupling capacitances. In some cases the coupling capacitances show large deviation. This happens because of the change in the adjacency of the nets causing some coupling capacitances to be large for certain sizes and negligible for others.

3. IMPACT OF PARASITICS

The impact of each of these capacitance categories on circuit performance was studied for each one of our benchmark circuits shown in Fig. 5. For each of these circuits we measured and compared performance attributes in the following situations of including or excluding the parasitic capacitances:

- without any parasitic capacitances
- with only the estimated diffusion parasitics (AD, PD, AS, PS).
- with all estimated intra-module caps but without any inter-module caps and
- with all the estimated capacitances.

The performances in each of the situations are compared against those obtained by following concrete layout generation and extraction for the same sizes. The statistics reported are for over 500 random validation runs and are presented in Fig. 7.

Analysis of circuit performance with no parasitic capacitances included shows the maximum performance deviation. This justifies the importance of accounting for layout parasitics during synthesis. Diffusion parasitic capacitances are responsible for a large percentage of performance degradation. Estimating them alone, brings down the performance deviation significantly in most cases. Inclusion of all other module parasitics helps model circuit performance more accurately. In most of the figures the fourth bar is hardly visible. Thus, modeling all parasitic capacitances reduces the performance deviation to negligible values. Some simulators do have a provision for estimating the diffusion parasitics at the pre-layout stage, however that estimation is not even close to accurate because it assumes that each transistor has one fold which is most often not true at the post-layout stage. Another noteworthy observation from the figures is that some performance parameters are more sensitive to parasitics as compared to others. For instance, for the Single Ended Opamp, phase margin is most sensitive to parasitics and for the Bandpass Filter, the bandwidth is extremely sensitive to parasitics. On the other hand, open loop ac gain for the opamp circuit is much less sensitive to parasitics and is dependent on the diffusion parasitics only as seen in Fig. 7. Similar results were observed for the twostage opamp.

4. SYNTHESIS METHODOLOGY

We propose a layout aware synthesis approach where the layout generation and extraction step is removed from the optimization loop. The layout effects are taken into account using parasitic models developed as shown in Fig. 8. It has already been illustrated that the proposed models have low mean error and provide a close approximation of the layout effects. The proposed flow is compared with layout-in-the-loop method where a PLG (same as that used to

Table 5: Deviation of Parasitics for Regenerative Comparator Topology

N1:N2	Mean Dev	Max Dev	Std Dev	N1:N2	Mean Dev	Max Dev	Std Dev
vdd:GND	0.003	0.018	0.007	m6g:GND	0.63	1.87	0.60
m7g:GND	0.70	1.906	0.64	vss:m1d	0.61	2.25	0.61
vss:m7g	0.98	4.46	1.07	vss:vinn	0.74	4.54	1.419
vss:m6g	1.09	4.64	0.99	vss:m0d	1.07	4.84	0.93
m6g:clk	1.51	5.55	2.1	vss:vinp	2.14	5.55	2.5
outn:GND	0.16	7.69	1.093	outp:GND	0.16	7.69	1.093
vss:outn	0.18	7.69	0.98	vss:outp	0.18	7.69	0.98
vss:GND	2.53	8.35	1.70	m7g:clk	2.53	9.09	3.32
m6g:m0d	2.17	9.71	2.33	m1d:GND	3.21	11.8	2.49
m0d:GND	3.27	14.2	2.62	m0d:m1d	3.44	14.7	3.21

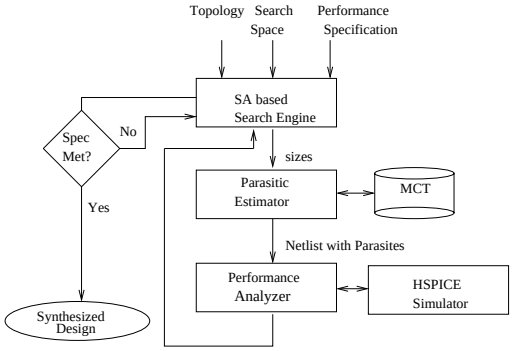

Figure 8: Synthesis Flow

build parasitic models) is used inside the optimization loop.

The simulated annealing based search engine determines device sizes in each iteration of the optimization loop. The module characterization table (MCT) is used to determine all parasitic capacitances. The output of the parasitic estimator is a netlist which closely resembles an extracted netlist. The performance estimator then simulates the netlist by invoking the HSPICE(v7.2) simulator and processes the simulation data to obtain the values of the desired performance parameters. This process is repeated until performance goals have been achieved.

5. EXPERIMENTAL RESULTS

Synthesis experiments were performed on our four benchmark circuits in TSMC 0.18μ technology. In Table 6, we present the synthesis results using the proposed approach (column 3). The fourth column represents the results obtained when the synthesized design obtained using the proposed technique is verified by layout generation, extraction and performance estimation. Comparing the results in columns three and four, its obvious that the estimated performance is very accurate. Synthesis results obtained using layout-in-the-loop optimization technique are shown in the fifth column. The proposed technique thus produces results comparable to those using PLGs.

A comparison of time spent for capturing the effect of layout using the proposed approach and layout-inclusive approach has been presented in table 7. A speedup of 88% or more is obtained for each benchmark. The proposed approach significantly reduces the time spent in including the effect of layout parasitics in the synthesis flow. Our synthesis methodology thus reduces the time with negligible loss of accuracy.

Table 6: Synthesis Results

Ckt	Performance Specification	Estimated	Actual	Layout In-Loop
TSO	Open loop Gain(dB) $\geq$ 40	40.6	40.6	40.1
	Phase Margin(°) $\geq$ 50	53.5	53.5	50.5
	3dB(MHz) $\geq$ 60	66.86	66.85	61.44
	UGF(MHz) $\geq$ 320	762	762	661
SEO	Open loop Gain(dB) $\geq$ 40	40.6	40.6	40.5
	Phase Margin(°) $\geq$ 50	52.64	52.62	53.28
	3dB(MHz) $\geq$ 8	8.48	8.53	9.85
	UGF(MHz) $\geq$ 320	618	620	786
BPF	Bandwidth(MHz) 45-70	46.64	46.65	45.77
	CenterFreq(MHz) $\geq$ 320	341	341	341.9
	Power Gain(dB) $\geq$ 5	5	5	5.25
	Q-Value $\geq$ 5.2	7.314	7.313	7.47
RC	Fall Time(ns) $\leq$ 0.41	0.39	0.39	0.4
	Rise Time(ns) $\leq$ 0.41	0.4	0.4	0.4
	SlewRate(1e10v/s)$\geq$1.4	1.4509	1.451	1.46

Table 7: Estimation vs Layout Gen and Extraction Time Per Iteration

Circuit	Estimation Time(s)	Layout Gen and Extraction Time(s)	Speedup (%)
SEO	0.087	2.785	96.85
TSO	0.405	3.721	89.11
RC	0.467	4.07	88.5
BPF	0.027	3.11	99.13

6. CONCLUSION

Models were developed for estimating various parasitic capacitances. The use of these models for performing layout aware synthesis was demonstrated. Synthesis results indicate that the proposed method achieves designs of similar quality as those obtained with layout-inclusive synthesis approaches while reducing the time spent in synthesis. From the study of the impact of parasitics on circuit performance, it can be concluded that some performance parameters are more sensitive to some parasitics as compared to others. Hence, the sensitivity of performance parameters to parasitics could be used to determine the crucial parasitics. The insignificant parasites can be ignored, thus reducing both the estimation and simulation time. In this work, only parasitic capacitances were

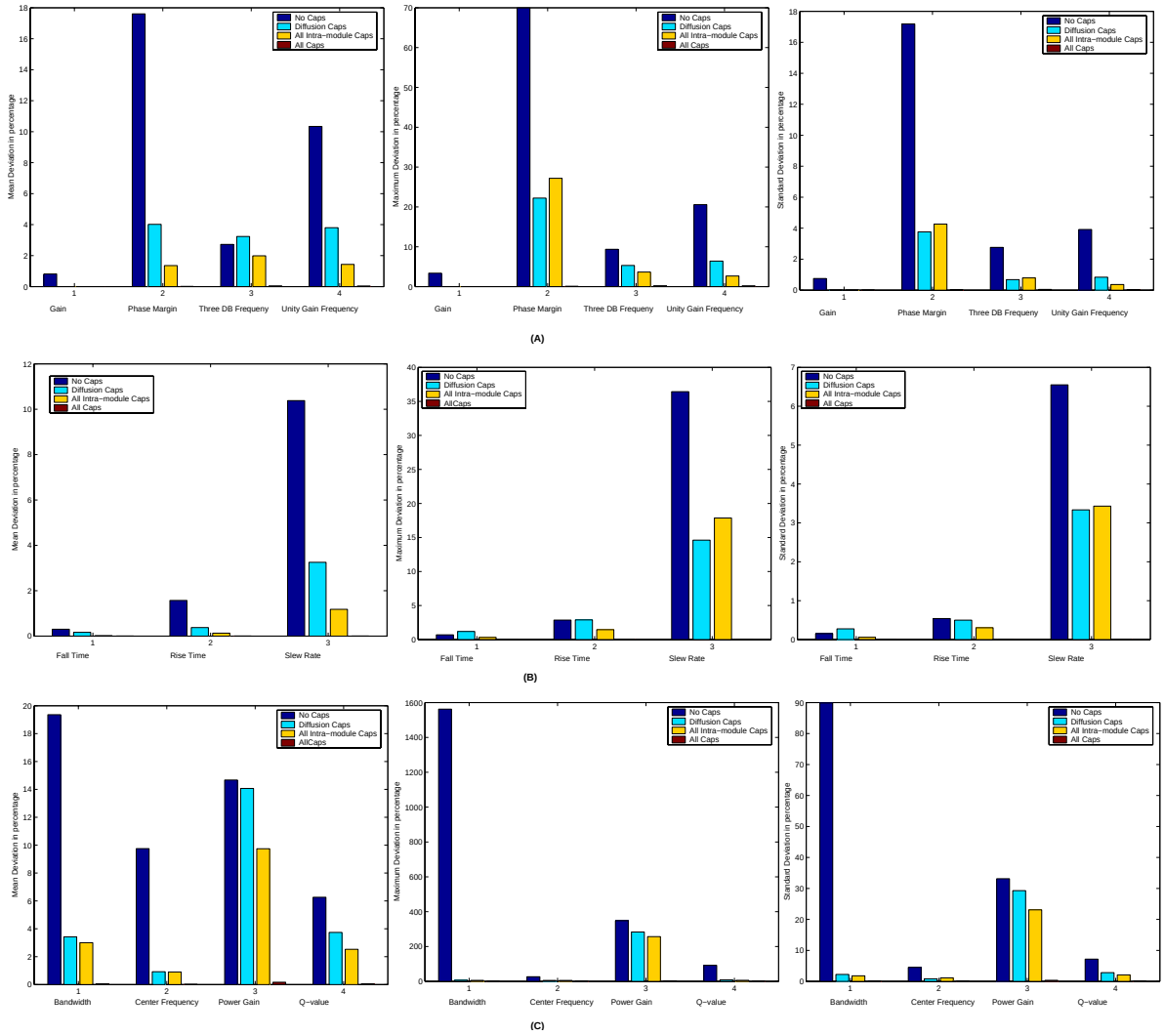

Figure 7: Performance Deviation for (A) Single Ended Opamp, (B) Regenerative Comparator, (C) Bandpass Filter

modeled which dominate the circuit behavior in the sub GHz range. To extend this approach for synthesis of RF circuits, we would have to consider resistive and inductive parasitics as well.

7. ACKNOWLEDGMENT

This work is a part of the SHARC (Synthesis of High Performance Analog and RF Circuits) tool-suite developed at University of Cincinnati.

8. REFERENCES

[1] H. Onodera et al. Operational amplifier compilation with performance optimization. *IEEE JSSC*, 25(2):466–473, April 1990.

[2] M. Dessouky, M. Louerat, J. Porte. Layout-oriented synthesis of high performance analog circuits. In *Proc. of the conference on Design Automation and Test in Europe*, pages 53–57, 2000.

[3] P. Vancorenland, G. Van der Plas, M. Steyaert, G. Gielen, W. Sansen. A layout-aware synthesis methodology for RF circuits. In *Proc. of the 2001 IEEE/ACM international conference on Computer-aided design*, pages 358–362, 2001.

[4] K. Choi, D.J. Allstot, and S. Kiaei. Parasitic-aware synthesis of RF CMOS switching power amplifiers. In *IEEE International Symposium on Circuits and Systems*, volume 1, pages 269–272, 2002.

[5] H. Sampath and R. Vemuri. MSL: A High-Level Language for Parameterized Analog and Mixed-Signal Layout Generators. In *Proc. of IFIP 12th International Conf. on VLSI*, 2003.

[6] G. Wolfe and R. Vemuri. Extraction and use of neural network models in automated synthesis of operational amplifiers. *IEEE Transactions on Computer-Aided Design*, 22(2):198–212, February 2003.

[7] S. S. Mohan, M. Hershenson, S. P. Boyd, and T. Lee. Simple accurate expressions for planar spiral inductors. *IEEE Journal of Solid-State Circuits*, 24(10):1419–1424, October 1999.

[8] A. Agarwal, H. Sampath, V. Yelamanchili, and R. Vemuri. Accurate Estimation of Parasitic Capacitances in Analog Circuits. In *Proc. of Conference on Design Automation and Test in Europe*, February 2004.