Hierarchical Modeling and Analysis of Process Variations in Deep Sub-micron Devices: AC Analysis

by

Jacob Flomenberg

Department of Electrical and Computer Engineering

4/27/05

Professor Sule Ozev, Advisor

A paper submitted in partial fulfillment of the requirements for graduation with departmental distinction for the degree of Bachelor of Science in the department of Electrical and Computer Engineering in the undergraduate school of

Duke University

2005

Acknowledgements

This work would not have been possible without the help of my research advisor, Dr. Sule Ozev. I would like to thank her for her sincere encouragement and support of my studies. Her constructive remarks and help throughout the project played an integral role in steering my research down the right path. I am also grateful to Fang Liu for aiding me in my technical analyses. Finally, I would like to acknowledge Devaka Viraj Yasaratne, who worked with me on implementing Java code relating to the project.

At this point in time I would also like to establish that Viraj and I did significant amounts of work on this project together before our research diverged. Both of us spent time developing the coding infrastructure to run sensitivity analyses and Monte Carlo simulations. Since then, we have split our research efforts as I now work exclusively on the AC analysis portion while he works on the DC analysis. Consequently, the focus of this paper will be the infrastructure and AC analysis. Viraj has made significant modifications to the code to facilitate his analysis so I will discuss the code in the state before he made any modifications on his own to it.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The semiconductor industry is progressing towards devices with smaller feature sizes, increased integration, and smaller time-to-market windows. As the feature size decreases, precise control over process parameters becomes more difficult, leading to larger percentage variations.[1]

As an example, consider the fact that the smallest possible deviation in the oxide thickness cannot be decreased below the average bonding length of SiO_2, 1.61Å[2]. For the oxide thickness reduction from 140Å in .5 µm technology[3] to 25Å in 90 nm technology[4], the minimum relative deviation went from 1.1% to 6.4%.

---

[1] ITRS, "International Technology Roadmap for Semiconductor," http://public.itrs.net/Files/2003ITRS/Home2003.htm, 2003.
[2] Y.P. Li, and W.Y. Ching, "Band Structures of All Polyscrystalline Forms of Silicon Dioxide," *Physics Review*, vol. B31, no. 4, pp. 2172-2179, Feb. 1985.
[3] American Microsystems Inc., "MOSIS Parametric Test Results," http://www.mosis.org/cgi-bin/cgiwrap/umosis/swp/params/ami-c5/t41c-params.txt, Apr. 2004.
[4] T. Devoivre, *et al.*, "Validated 90nm CMOS Technology Platform with Low-k Copper Interconnects for Advanced Systems-on-Chip (SoC)," in *IEEE International Workshop on Memory Technology, Design and Testing, 2002*, Jul. 2002, pp. 157-162.

The SIA roadmap already acknowledges that the statistical variation in parameters such as threshold voltage and oxide thickness may become intolerably high.[5] Thus it is clear that process variations will play an increasingly large role in device performance.

While both digital and analog circuits are affected by such deviations, digital circuits may mask some of this behavior as some deviations do not induce changes at the output of the circuit. Thus, the fact that analog circuits reflect all process variations because of their continuous range of values makes them better candidates for study.

Comparing and quantifying parametric variations have a myriad of uses in VLSI design and testing. The incorporation of process variations into design flow from early on in the development process can help identify yield, boundaries for fault detection, and provide insight into the feasibility of a design.

Traditional evaluation of statistical variations in component values on a response function is known as the statistical tolerance analysis problem. There have been numerous methodologies developed for different application domains. However, the methods that are applicable to variance analysis are lacking in terms of the ability to trade off accuracy and complexity. Highly accurate techniques such as Taguchi's method use deterministic sampling to provide results but take $O(3^n)$ time to execute.[6] Similarly, methods that take only a fraction of this time to execute, like worst case min-max analyses based on weighted sensitivities, provide quick results but often at the expense of setting unnecessarily stringent tolerance bounds.[7]

---

[5] ITRS, "International Technology Roadmap for Semiconductor," http://public.itrs.net/Files/2003ITRS/Home2003.htm, 2003.

[6] G. Taguchi, "Performance Analysis Design," *International of Production Research*, vol. 16, pp. 521-530, 1978.

[7] M. Tian and R.-J. Shi. Worst case tolerance analysis of linear analog circuits using sensitivity bands. *IEEE TCAS-I*, 57(8):1138–1145, August 2000.

My fellow researchers have developed a theoretical variance analysis technique that promotes data re-use. This method provides both accuracy and computational efficiency compared to prior approaches. Furthermore, by setting certain performance criteria the method can meet a whole host of points in the accuracy and efficiency spectrum.

My role is to help develop a hierarchical model to implement the proposed method so that we may further characterize its performance. The main focus of my work will be to facilitate a quantitative comparison between this method and prior techniques.

To this end, a coding infrastructure was developed in order to run circuit simulations and collect data. The results were then processed and re-incorporated where necessary. Additionally, analytical models were derived for AC analysis, and practical issues for the implementation of the proposed variance analysis technique were addressed.

Chapter 2

Code Infrastructure

It was necessary to develop a software infrastructure in order to run circuit simulations and collect data. The infrastructure was required to perform both sensitivity analyses and Monte Carlo simulations to form a basis of comparison for the proposed variance analysis technique. Sensitivity analysis was to be performed using a first order Taylor series approximations and the Monte Carlo functionality had to provide the ability to turn on and off variability for each parameter.

The first step was to develop software capable of executing system calls so that it could interact with HSPICE, software developed for circuit behavior simulation. To this end, research was conducted to determine how to execute system calls so that HPSICE could be invoked from code in the software being developed. The first few iterations of the system call functionality were developed using C.

The software had to be able to alter HSPICE input parameters, parse HPSICE output files, and to store the output information for later processing. Initially, C was

chosen for this task and both parsing and system calls took place in C code. However, it became clear that Java was better suited to the task of parsing and storing strings as it has a suite of built-in functionality.

So the system call functionality of the C code was expanded to not only call HSPICE but also to run small Java programs to parse and process output. Once it was evident that the bulk of coding needed to be done in Java, the C shell script was scrapped and system call functionality was implemented in Java. Over the course of the first semester a full suite of Java code was developed to facilitate both sensitivity analysis and Monte Carlo circuit simulations to be used as the basis for comparisons of the proposed variance analysis technique to prior approaches. All programs mentioned can be found in the appendix.

The Monte Carlo circuit simulation is implemented using a java shell program named MonteCarlo. The program calls a number of different programs. It first calls HspiceFileGrabber on the input file. HSpiceFileGrabber parses an input file specified a propriety HSPICE-like format into a standard HSPICE format. It also stores what parameters will be allowed to vary during the circuit simulations. The inclusion of this information in the input file is the major difference between the actual input file and a normal HSPICE circuit description. The MonteCarlo program then calls Pickvals to generate the random circuit descriptions. Pickvals goes through each input parameter and varies those parameters set to be variable to create the number of random circuit descriptions specified by the program parameter NUMCIRCUITS. These circuit descriptions are stored in a file specified by the input variable myInputFile and the names of the parameters are stored in a file specified by the input variable PARAMFILE.

Finally, HspiceFileCreator is called to simulate the circuits. HspiceFileCreator uses the information in myInputFile and PARAMFILE to simulate each circuit description. After each simulation, GrabSpice is called to grab all the outputs of that run and write it to "randout.out". After all the simulation runs are complete GrabParam is called to collect output values of each parameter.

Sensitivity analysis is peformed by calling calcSens01. CalcSens01 first calls HspiceFileGrabber to parse the input. The circuit information is stored such that each parameter selected for sensitivity analysis can be varied while holding the rest at their nominal values. The next program called is HspiceFileCreator (a different version than the one used for Monte Carlo). This program calls CircuitBuilder to generate circuit descriptions with each of the input parameters varied by a given percentage. HspiceFileCreator uses the information in myInputFile and PARAMFILE to simulate the circuit descriptions created by CircuitBuilder. Then GrabSpice is called to grab all the outputs of that run and write it to file. Finally, CalcSensitivity is called to calculate the sensitivity to each input parameter to all output parameters.

Additionally, the program GetFreq is used to calculate AC characteristics such as unity gain, gain bandwith product, and 3dB frequency from HSPICE output. This can be used in conjunction with either the sensitivity or Monte Carlo packages to include AC characteristics in the output parameters examined.

With the code infrastructure complete, simulations were run to verify that the code behaved properly and the results generated were within expected tolerances. Results generated from this infrastructure are discussed further in the experimental results section.

The following figure summarizes the functionality of the suite:

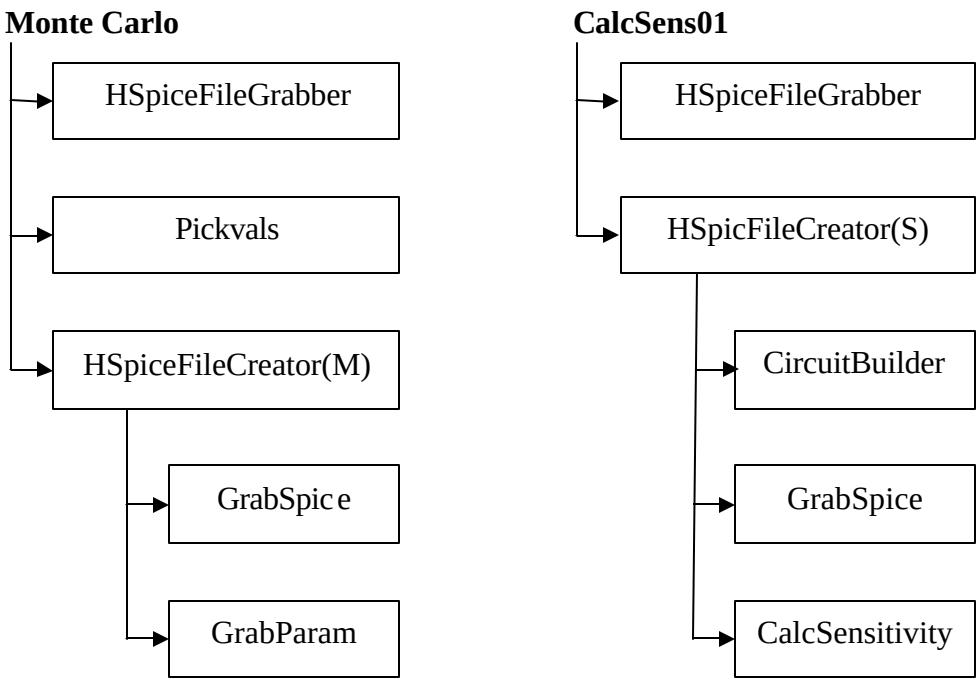

Figure 2.1: Diagram of Infrastructure Behavior

Chapter 3

AC Analysis

The circuit chosen for analysis was a differential pair with a current mirror load. In order to assess the validity of the proposed variance analysis technique it was necessary to assess its performance in the frequency domain. To this end, an AC analysis of several current mirrors of increasing complexity was undertaken.

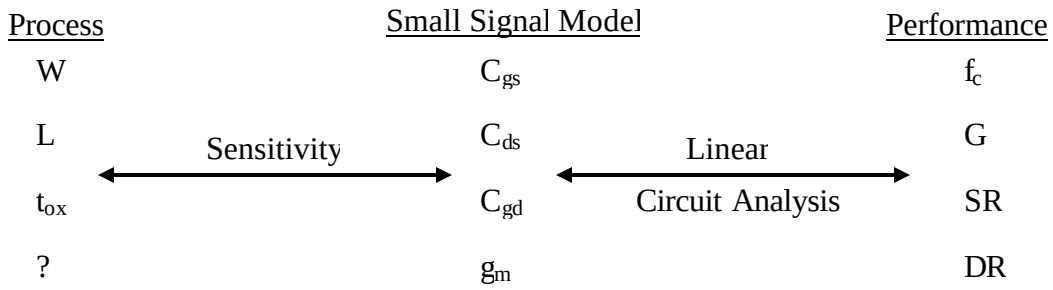

Figure 3.1: Small signal model framework

A representative small signal model framework is shown above. The process level parameters are related to the small signal model parameters based on sensitivity

analysis results. While the performance model parameters can be related to small signal parameters using analogous sensitivity-based techniques, the study first focuses on determining whether or not this relationship can be captured by analytically derived formulas.

Based on data collected, the following parameters were chosen as the focus of the study because of their high sensitivities:

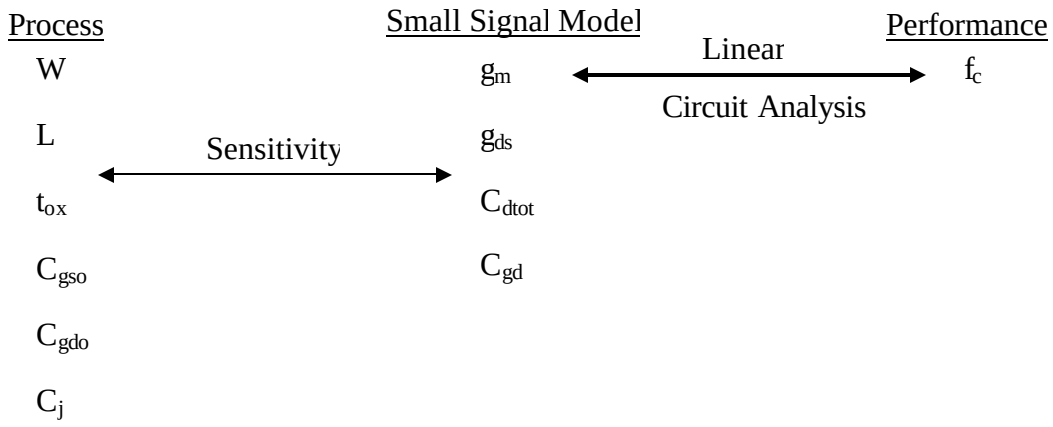

Figure 3.2: Small signal model analyzed

The first differential pair circuit examined is shown below. MOSFETs 1 and 2 form a current mirror, MOSFET 3 biases the current mirror and resistor 4 balances the load seen by the current mirror.

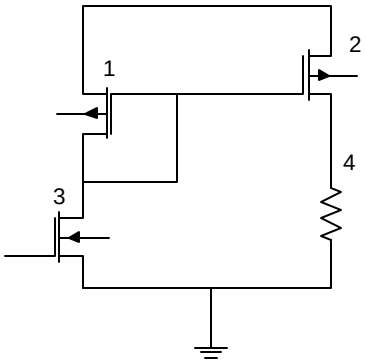

Figure 3.3: Circuit diagram of simple differential pair

The next 3 figures detail the small signal analysis performed. The first simply shows the small signal model of the current mirror.

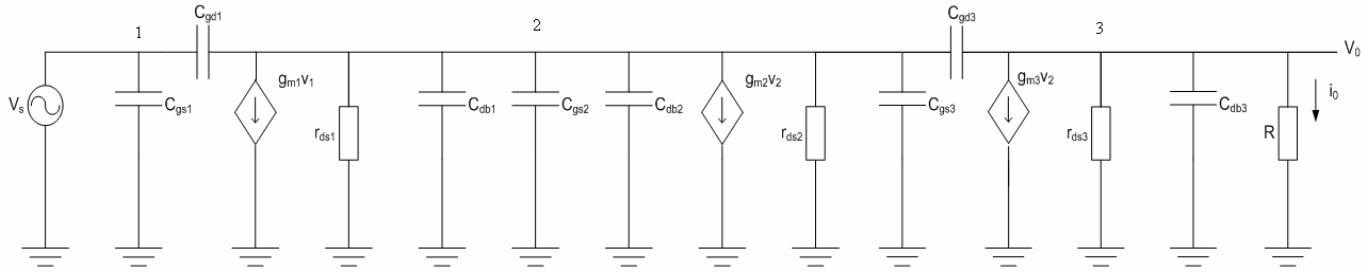

Figure 3.4: Simple differential pair small signal model

The following substitutions simply the circuit substantially:

$$r_1 = r_{ds1} + r_{ds2}$$
$$r_2 = r_{ds3}$$
$$r_3 = R$$
$$C_1 = C_{gs1}$$
$$C_2 = C_{gd1}$$
$$C_3 = C_{db1} + C_{gs2} + C_{db2} + C_{gs3}$$
$$C_4 = C_{gd3}$$
$$C_5 = C_{db3}$$

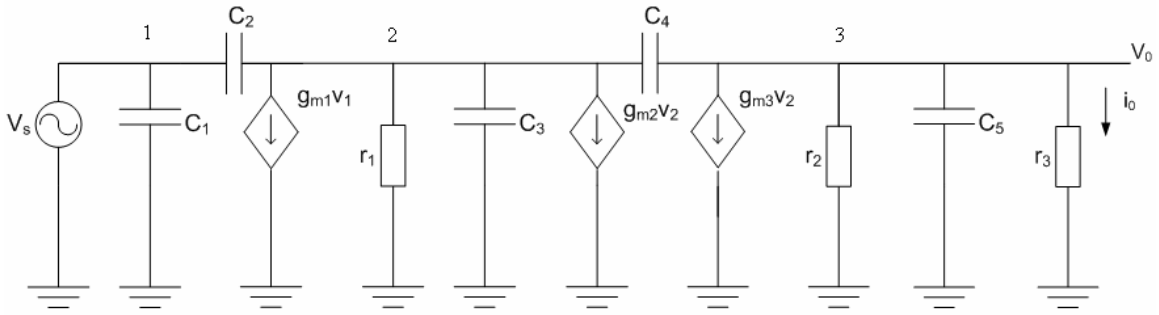

Figure 3.5: Simple differential pair reduced small signal model

From this model, the gain and input current were derived:

$$g_{m1}v_1 + \frac{v_2}{r_1} + g_{m2}v_2 = 0$$

$$g_{m1}v_1 = -(g_1 + g_{m2})v_2$$

$$v_2 = \frac{-g_{m1}v_1}{(g_1 + g_{m2})}$$

$$g_{m3}v_2 + \frac{v_3}{r_2} + \frac{v_3}{r_3} = 0$$

$$g_{m3}v_2 = -(g_3 + g_2)v_3$$

$$v_3 = \frac{g_{m3}}{(g_3 + g_2)} \times \frac{g_{m1}}{(g_1 + g_{m2})} \times v_s$$

$$i_0 = \frac{g_{m1} \times g_{m3} \times g_3}{(g_3 + g_{m2})(g_1 + g_{m2})}$$

$$A = \frac{g_{m1} \times g_{m3}}{(g_3 + g_{m2})(g_1 + g_{m2})}$$

Finally, Miller's approximation was employed to derive pole information for the circuit.

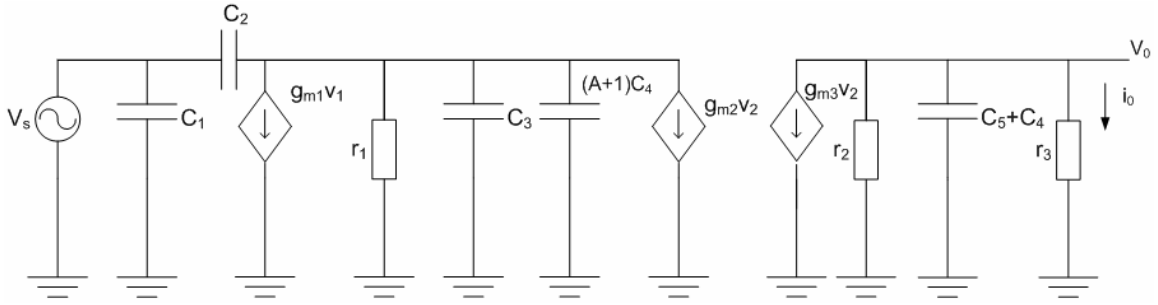

Figure 3.6: Simple differential pair reduced small signal model with Miller's approximation

$$[g_3 + g_2 + s(C_4 + C_5)]v_3 + g_{m3}v_2 = 0$$

$$[g_{m2} + g_1 + s(C_3 + (A+1)C_4 + C_2)]v_2 + (g_{m1} - sC_2)v_1 = 0$$

$$w_2 = \frac{g_3 + g_2}{C_4 + C_5}$$

$$w_1 = \frac{g_{m2} + g_1}{C_3 + C_2 + (A+1)C_4} \quad \longrightarrow \text{dominant pole} \rightarrow f_c$$

While this exercise was useful in an academic sense, the simple differential pair small signal model does not provide enough information to accurately model the much more complex HSPICE description of the differential pair..

The complete differential pair as described in the HSPICE circuit description under study is shown below:

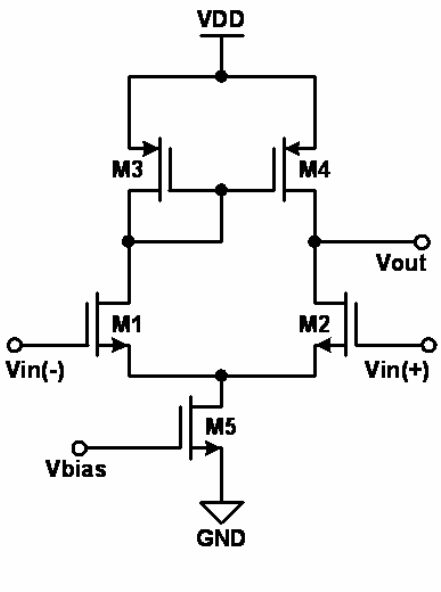

Figure 3.7: Circuit diagram of HSPICE differential pair

The small signal model for this circuit consists of 18 parameters over 5 separate nodes. This level of complexity prevents the node voltage method from being applied by hand. The node voltage equations were entered into Mathematica but it was unable to isolate any poles in a manner that would be useful for further interpretation.

Thus, a HSPICE circuit description of the AC model was coded and simulated. A sensitivity analysis was then conducted to determine which parameter variations generated the largest change in cutoff frequency.

Paramater Modified	CutoffFrequency	Sensitivity	Normalized Sensitivity
NOMINAL	3.1735E+03	-	
cgs1	3.1737E+03	6.3022E-05	2.00
cgd1	3.1735E+03	0.0000E+00	0.00
rgds1	3.1733E+03	6.3022E-05	2.00
rgds4	3.1734E+03	3.1511E-05	1.00
c1	3.1388E+03	1.0934E-02	347.00
cgd4	3.0944E+03	2.4925E-02	791.00
c3	2.7552E+03	1.3181E-01	4183.00
rgds2	2.6694E+03	1.5885E-01	5041.00
cgd2	3.1379E+03	1.1218E-02	356.00
cgs2	3.1726E+03	2.8360E-04	9.00
rgds5	3.1735E+03	0.0000E+00	0.00
c2	3.1737E+03	6.3022E-05	2.00
ccv1	3.4494E+03	8.6939E-02	2759.00
ccv2	2.9856E+03	5.9209E-02	1879.00
ccv3	3.4180E+03	7.7044E-02	2445.00
ccv4	2.9370E+03	7.4523E-02	2365.00
ccv5	3.1253E+03	1.5188E-02	482.00

Table 3.1: Sensitivity analysis results for HSPICE differential pair small signal model

From this data, the 6 most sensitive parameters were selected for further analysis. All others were treated as open circuits. This yielded the following circuit diagram:

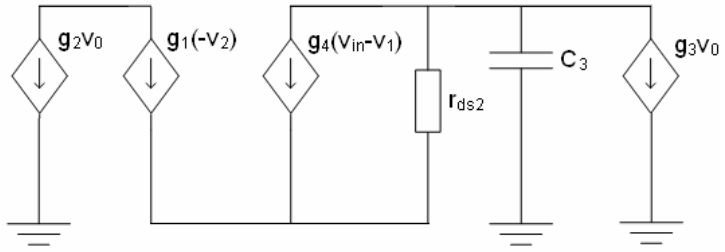

Figure 3.8: Circuit diagram of HSPICE differential pair small signal model

The node voltage method of analysis was used to derive the following analytical relation for gain:

$$G = \frac{g_4 + \dfrac{g_4 \times \left(\dfrac{g_1 \times g_3}{g_2} - g_4 - g_{ds2} \right)}{g_1 + g_4 + g_{ds2}}}{s \times C_3 + g_{ds2} + \dfrac{\left(\dfrac{g_1 \times g_3}{g_2} - g_4 - g_{ds2} \right) \times g_{ds2}}{g_1 + g_4 + g_{ds2}}}$$

From this relation, the cutoff frequency was derived to be:

$$f_c = f_{c0} + \frac{\sqrt{\left(g_{ds2} + \frac{\left(\frac{g_1 \times g_3}{g_2} - g_4 - g_{ds2} \right) \times g_{ds2}}{g_1 + g_4 + g_{ds2}} \right)^2}}{2 \times p C_3^2}$$

Unfortunately, while this relation captures the sensitivity of the cutoff frequency, it requires knowledge of the nominal offset. In order to calculate the variance one needs to differentiate this equation with respect to each of the small signal parameters in order to derive sensitivity relations. Then the sensitivity data can be utilized in the last step of the construct discussed in the paragraphs which follow.

The complexity of analysis itself and the practical issues raised by it makes it clear that analytical relations can not efficiently be used to relate small signal parameters to higher level parameters. Furthermore, the analytic expressions for the level 49 HSPICE differential pair circuit are quite complicated and could not be investigated under the time constraints imposed.

Thus, in order to determine the accuracy of the model, a covariance matrices based method was employed. The method is based on the theorem for a general hierarchy. It incorporates linearization using first order Taylor approximations and hierarchical implementation with matrix multiplication to relate the low level parameters to the high level parameters.

From the following equation we can derive the basics of the model:

$$p = p_{nom} + [S][\Delta x]$$
$$\Delta p = [S][\Delta x]$$

Here, p denotes the high level parameter in question, S denotes a sensitivity matrix, and ? x denotes a square matrix containing the process parameter variances along the diagonal. The covariance matrices approach simply says that the following construct can be used to hierarchically determine covariance matrices:

$$\Sigma_p = [S][\Sigma_x][S^T]$$[8]

In order to determine the cutoff frequency variation, it was necessary to iterate through two levels of hierarchy. First, the sensitivity of small signal model parameters to the process parameters was determined. With this data, the small signal model parameter covariance matrix was calculated.

$$\Sigma_{SS} = \left[S^{SS}_{process}\right]\left[\Sigma_{process}\right]\left[S^{SS}_{process}\right]^T$$

Then, this data was used along with the sensitivities of the small signal parameters to the cutoff frequency to find the cutoff frequency covariance matrix.

$$\Sigma_{f_c} = \left[S^{f_c}_{SS}\right]\left[\Sigma_{SS}\right]\left[S^{f_c}_{SS}\right]^T$$
$$s_{f_c} = \sqrt{\Sigma_{f_c}}$$

This result is a 1 X 1 matrix which denotes the cutoff frequency variance. The cutoff frequency standard deviation is simply he square root of the variance.

Monte Carlo simulations of the differential pair circuit yield $s_{f_c} = 1.42E6$. The result of the covariance matrix analysis is several orders of magnitude smaller than this result. It was recently discovered that circuit versioning issues amongst different investigators created faulty data that mixes the behavior of slightly different circuits. Measures are currently being taken to correct this problem and then the analysis will be redone. The next section details those comparisons that are both correct and complete.

[8] Narayan Giri, Multivariate Statistical Analysis, pg. 59, 1995.

Chapter 4

Results of Loop Tracing Variance Analysis

With the data collected from the Monte Carlo simulations, sensitivity analyses, and the AC analysis, enough data was collected to assess the performance of the proposed variance analysis technique.

The proposed technique was implemented by conducting a variance analysis assuming independence among variables at a given level. Correlation correction terms were then computed by backtracking to lower levels. At each level, only variance, correlation corrections, and relations among parameters at adjacent levels are stored. This construct allows for information re-use as changes in the circuit description only necessitate updating those variables affected by the changes. Further details regarding the implementation are discussed in more depth in "Hierarchical variance analysis for analog circuits based on graph modeling and correlation loop tracing," which can be found in the appendix.

In order to assess the validity of the variance analysis technique, a four level highly non-linear hierarchical construct was developed:

$$F_1 = X_7X_2 + X_2^2 + X_2X_3X_6/X_4$$
$$F_2 = X_1^2 + X_3X_6 + X_4^2$$
$$F_3 = X_2X_6 + X_3X_4 + X_8^2$$
$$F_4 = X_1X_8 + X_4^2 + X_5X_7$$
$$F_5 = X_1^2 + X_7^2 + X_4^3/X_5$$
$$G_1 = F_1F_2 + F_3^2$$
$$G_2 = F_2^2 + F_4^2$$
$$G_2 = F_2^2 + F_4^2$$
$$H = G_1^2 + G_2G_3$$

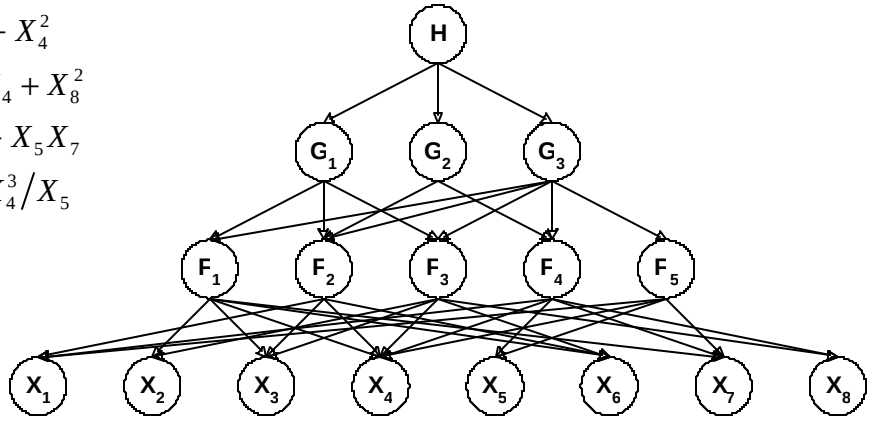

Figure 4.1: Non-linear hierarchical construct

Numerous different approaches were used to calculate the standard deviations for each parameter. The following figure compares the results of the approaches applied. The basis of comparison is a 50K Monte Carlo simulation.

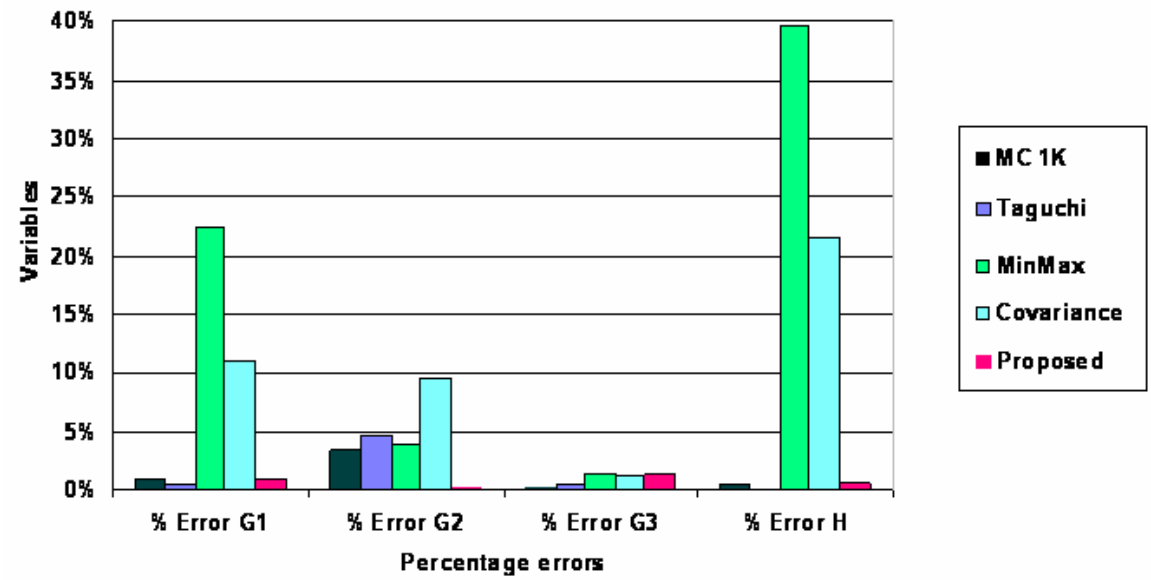

Figure 4.2: Comparison of percentage errors in standard deviation for various approaches

Analysis Method	Computational Time (s)
Monte Carlo 1K	46.5
Taguchi	328.3
Min-max	0.5
Covariance	0.8
Proposed	0.8

Table 4.1: Computation time for approaches applied

The results suggest that the min-max and covariance-based approaches are of limited use in conducting hierarchical analysis because of their poor accuracy. Furthermore, the proposed method does track with the results and takes only a fraction of the time that Taguchi's method or a 1K Monte Carlo simulation takes.

The hierarchical construct was then applied to the differential pair circuit.

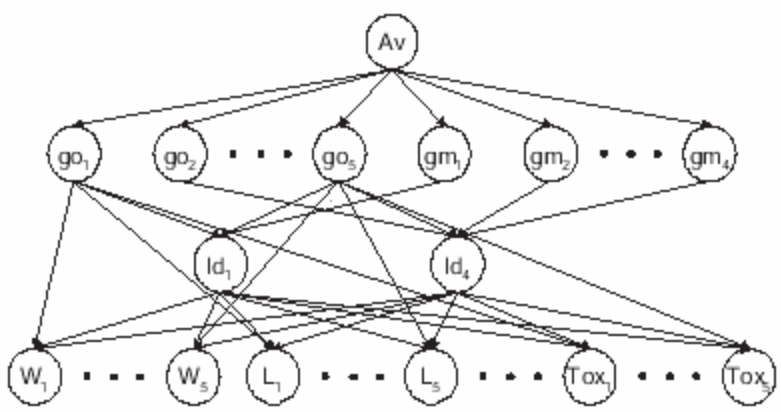

Figure 4.3: Hierarchical construct for differential amplifier

Table 2 shows the accuracy and computation time for the application of the min-max, covariance, and proposed approaches. A 12K Monte Carlo simulation is used as the basis of comparison. Taguchi's method is not included in this analysis as the sample size with 30 process level parameters is computationally intractable.

σ	Monte Carlo 12K	Min-max	Covariance	Proposed
% ID1	-	681.1	-0.3	2.0
% ID4	-	698.9	-0.3	2.0
% gm1	-	1641.3	470.0	3.1
% gm2	-	1639.1	468.5	3.6
% gm3	-	1623.8	466.5	2.6
% gm4	-	1610.8	459.8	1.5
% go1	-	695.8	6.9	2.3
% go2	-	651.9	1.0	-3.4
% go3	-	695.0	6.8	2.1
% go4	-	625.2	-2.6	-6.7
% go5	-	574.6	-7.1	3.0
% Av	-	1585.4	414.3	9.4
Time (s)	**$5.0\ 10^4$**	**1.7**	**1.7**	**2.4**

Table 4.2: Accuracy and computation time for differential pair circuit variance analysis

The proposed method tracks the Monte Carlo results within 10% and takes only a fraction of the time. While the min-max method is highly inaccurate, the covariance method appears to be hit or miss. This most likely has to do with the fact that the covariance method makes some assumptions of independence and they only hold for the cases in which the method produces accurate results.

Chapter 5

Conclusion and Future Work

As circuit complexity and the role of process variability become more important, there will be an ever increasing need for efficient and accurate methods to compute variability in performance parameters. The efficiency of the state of the art can be improved upon by capitalizing on the hierarchical nature of VLSI systems. Similarly, the accuracy of the state of the art can be improved upon by properly taking the statistical distributions and correlations among circuit variables into account.

In this study, an infrastructure was developed to facilitate the comparison of a novel variance analysis technique with prior approaches. Additionally, the frequency domain response of the circuit being studied was analyzed to assess the feasibility of deriving analytic relations to relate the small signal model parameters to the performance parameters.

During this process it was discovered that node voltage analysis becomes very complex for large circuits and should not be used in an efficient algorithm for calculating

variance. Furthermore, it was determined that small-scale independent blocks allow for a sensitivity based approach to be employed in calculating the variance of performance parameters.

The comparison results indicate that the proposed hierarchical variance analysis methodology provides almost the same computational efficiency as a simple min-max and fixed-covariance approaches but provide much higher accuracy.

The next step is to extend the infrastructure from a proof of concept to facilitate the analysis of much larger analog circuits. Ultimately, through understanding the impact of process variations, one can pave the way for designing deep sub-micron devices which are robust to the imperfections in the manufacturing process.

Appendix

The following paper is included in the appendix and provides an overview of the proposed variance analysis method:

Liu F, Flomenberg J, Yasaratne D, Ozev S. “Hierarchical variance analysis for analog circuits based on graph modeling and correlation loop tracing”, submitted to *IEEE Design, Automation and Test in Europe*, 2005