

# Ensuring Correctness of Analog Circuits in Presence of Noise and Process Variations Using Pattern Matching

Rajeev Narayanan, Mohamed H. Zaki and Sofiène Tahar

Dept. of Electrical and Computer Engineering, Concordia University, Montreal, Quebec, Canada

Email: {r\_naraya, mzaki, tahar}@ece.concordia.ca

**Abstract**—This paper relies on the longest closest subsequence (LCSS), a variant of the longest common subsequence (LCS), to account for noise and process variations inherited by analog circuits. The idea is to use stochastic differential equations (SDE) to model the design and integrate device variation due to the  $0.18\mu\text{m}$  fabrication process in a MATLAB simulation environment. LCSS is used to find the longest and closest subsequence that matches with the subsequence of an ideal circuit. We illustrate the proposed approach on a Colpitts oscillator circuit. Advantages of the proposed methods are robustness and flexibility to account for wide range of variations.

## I. INTRODUCTION

Today's analog circuit verification techniques will have to incorporate a broader set of constraints (physical and functional) early in the design cycle in order to meet the circuit performance. Given the widening gap between the complexity of analog designs and the maturity of computer-aided design (CAD) tools [5], a combination of traditional and new verification strategies are needed to mitigate several effects such as noise [6] and process variation [2].

Traditionally, designers use a combination of statistical modeling and Monte Carlo simulation to verify noise and process variation in analog designs right from circuit level, through behavioral level [5]. But, monitoring techniques at circuit level often rely on manual (visual) inspection of the simulation results, thereby slowing down the design and verification process. More recently, monitors [7] have been incorporated in the verification loop by modeling the design with thermal noise in additive form using SDEs [3] and then combine device variation due to the  $0.18\mu\text{m}$  fabrication process in an assertion based run-time verification. Unfortunately, the limitation is the complexity involved in defining such monitors. Techniques based on formal methods, such as statistical model checking [4] has been successfully used to verify the saturation property in simple analog circuits, such as a third-order  $\Delta\Sigma$  modulator. Such model checking approaches are still in their infancy and can easily run into state-space explosion for complex circuits. Also, due to the complexity of generating formal models and the computational overhead of the algorithms used, success has been mainly limited by scalability issues.

This paper tries to address some of the shortcoming of the above approaches, by automatically ensuring the correctness of analog circuits in presence of noise and process variation

using the concept of pattern matching. For instance, given a circuit that has been simulated by  $N$  different designers across different design centers and with the same set of constraints, generating  $N$  different outputs that vary slightly with the ideal circuit output (Figure 1), we address the question of “*how to decide on the acceptance/rejection of those circuits?*” Also, for different Monte-Carlo trials, if the transient simulation output of the non-ideal circuit follows the output of an ideal circuit for say 99.0% and violates just 1.0% of the simulation time, the question remains: “*Does the designer have to reject the circuit entirely?*” To better answer the above questions, we need techniques that can provide a better assessment and reasoning for a given analog circuit.

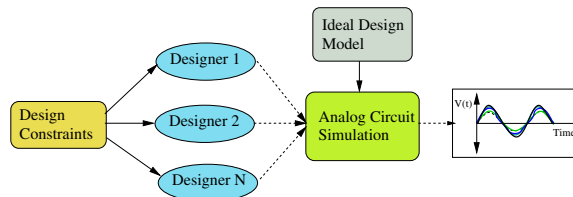


Fig. 1. Analog Verification.

Longest Common Subsequence (LCS) is a pattern matching algorithm that finds its applications in computational biology, chip layout design, and so on. In DNA matching, the idea is to find the longest subsequence common to all sequences in a set of sequences [8]. As opposed to the traditional approach of comparing the output of the design to its specification value, we can extend the LCS algorithm to estimate, in terms of percentage, the exact (100%) or “*closely*” matched simulated output relative to the ideal circuit output. By doing so, instead of blindly rejecting the circuit that violates the specification, designers will have more information during the evaluation and hence can make viable decisions. In reality, it is difficult to find a one-to-one mapping between two analog sequences and hence, designers have to define acceptable tolerance range for the output as a part of the specification in order to determine a “*closely*” matching sequence.

In this paper, we define to the best of our knowledge, the first longest closest subsequence (LCSS) algorithm for the verification of analog designs in the presence of noise and process variation. We propose to use stochastic differential equation (SDE) [3] to model the designs and then integrate the device variation due to the  $0.18\mu\text{m}$  [1] fabrication process in a MATLAB simulation environment. Then, we use LCSS to find the longest and closest set of subsequences that match with

the subsequence of an ideal circuit. Our approach is illustrated on a Colpitts oscillator circuit.

## II. LONGEST CLOSEST SUBSEQUENCE (LCSS)

Given two sequences of analog circuit output values  $X = \{X_1, X_2, \dots, X_m\}$  and  $Y = \{Y_1, Y_2, \dots, Y_n\}$ , then  $\exists Z = \{Z_1, Z_2, \dots, Z_k\}$ , an increasing maximum-length common subsequence, if  $Z$  is a subsequence of both  $X$  and  $Y$ . Here, we choose  $X$  to be the output sequence of an ideal circuit and  $Y$  is the output of the non-ideal circuit. To extend the LCS theorem [8] for analog circuits, we need to define a tolerance parameter  $p$  that describes the allowable boundary conditions for the sequence  $Y$  with respect to  $X$ . Tolerance level can be defined as part of a specification or can be considered as an equivalent of the confidence interval in a statistical simulation.

*Property 1:* Let  $X = \{X_1, X_2, \dots, X_n\}$  and  $Y = \{Y_1, Y_2, \dots, Y_m\}$  be sequences, and let  $Z = \{Z_1, Z_2, \dots, Z_k\}$  be any LCSS of  $X$  and  $Y$ . Then, for a given tolerance level  $p$ ,

- (1) If  $Y_n \leq (1+p)X_m$  and  $Y_n \geq (1-p)X_m$ , then  $Z_k$  is an LCSS of  $X_m$  and  $Y_n$ .
- (2) If  $X_m \neq Y_n$ , then  $Z_k \neq X_m$ , then  $Z$  is an LCSS of  $X_{m-1}$  and  $Y$ .
- (3) If  $X_m \neq Y_n$ , then  $Z_k \neq Y_n$ , then  $Z$  is an LCSS of  $X$  and  $Y_{n-1}$ .

A brute-force method to compute  $\text{LCSS}(X, Y)$  involves computing all subsequences of  $X$ , checking if they are subsequences of  $Y$  and be the longest. For a given sequence ( $X$  and  $Y$ ) of length  $m$  and  $n$ , respectively, we have  $2^m$  subsequences of  $X$  and it takes  $O(n \cdot 2^m)$  to compute the LCSS. As we can see from Property 1, there are overlapping subproblems in finding the LCSS of  $X$  and  $Y_{n-1}$  and the LCSS  $X_{m-1}$  and  $Y$ , hence, it is efficient to implement the above property recursively with the computation time of  $O(m+n)$ . The recursive solution to the LCSS problem is based on finding an intermediate length  $C[i, j]$  of  $\text{LCSS}(X, Y)$ , where  $i$  and  $j$  represent the  $i^{\text{th}}$  and  $j^{\text{th}}$  positions of  $X$  and  $Y$ , respectively. If either  $i = 0$  or  $j = 0$ , one of the sequences has length zero, so the LCSS has length zero as defined in Property 2.

*Property 2:* Require:  $i, j \neq 0$

$$C[i, j] = \begin{cases} C[i-1, j-1] + 1; & (1-p)X_m \leq Y_n \leq (1+p)X_m \\ \max\{C[i, j-1], C[i-1, j]\}; & \text{otherwise} \end{cases}$$

The implementation of the Properties 1 and 2 is described in Algorithm 1.  $B_x[i, j]$  and  $B_y[i, j]$  (lines 13 and 14) point to the table entry corresponding to the solution when computing  $C[i, j]$ . The entries in the matrix table can be described as follows: First, we define an intermediate LCSS  $C[i, j]$  and initialize it to zeros (lines 3-8). If the sequence  $Y_{ij}$  does not match with  $X_i$ , then based on the comparison between  $C[i-1, j]$  and  $C[i, j-1]$  (lines 15-23) we determine the corresponding value in the table. If sequence  $Y_{ij}$  does match with  $X_{ij}$ , then  $C[i, j]$  is computed by adding "1" to the value in the position  $C[i-1, j-1]$  (lines 12-14). The algorithm continues until we reach the end of the sequence, and it returns the matrix table  $C[i, j]$  (line 27).

Now, using the table  $C[i, j]$  and the position matrices  $B_x[i, j]$  and  $B_y[i, j]$  we can trace back the path to determine

---

## Algorithm 1 LCSS Algorithm

---

```

Require:  $X, Y, p$ 
1:  $m \leftarrow \text{Length}[X]$ 
2:  $n \leftarrow \text{Length}[Y]$ 
3: for  $i \leftarrow 1$  to  $m$  do
4:    $C[i, 0] \leftarrow 0$ 
5: end for
6: for  $j \leftarrow 1$  to  $n$  do
7:    $C[0, j] \leftarrow 0$ 
8: end for
9: for  $i \leftarrow 1$  to  $m$  do
10:  for  $j \leftarrow 1$  to  $n$  do
11:   if  $(Y_n \leq (1+p)X_m) \ \&\& \ (Y_n \geq (1-p)X_m)$  then
12:      $C[i, j] \leftarrow C[i-1, j-1] + 1$ 
13:      $B_x[i, j] \leftarrow i-1;$ 
14:      $B_y[i, j] \leftarrow j-1;$ 
15:   else if  $(C[i-1, j] \geq C[i, j-1])$  then
16:      $C[i, j] \leftarrow C[i-1, j]$ 
17:      $B_x[i, j] \leftarrow i-1;$ 
18:      $B_y[i, j] \leftarrow j;$ 
19:   else
20:      $C[i, j] \leftarrow C[i, j-1]$ 
21:      $B_x[i, j] \leftarrow i;$ 
22:      $B_y[i, j] \leftarrow j-1;$ 
23:   end if
24: end for
25: end for
26:  $Z \leftarrow [B_x, B_y]$ 
27: return  $C, Z$ 

```

---

the longest closest sequence between the two analog output sequences<sup>1</sup>. We simply begin by defining the path with  $B_x$  and  $B_y$  and the elements of the LCSS are encountered in reverse order by this method.

Though, a closely matched sequence between  $X$  and  $Y$  was found, sometimes, we may come across a false matching sequence. For instance, in case of oscillators, we may come across false oscillation due to noise and process variation. In that case, if the tolerance parameter is large, then the LCSS algorithm might consider a matching case. Hence, the LCSS derived in this case is considered to be *false positive*. One way to describe this is by means of a *confusion matrix* [8] in order to identify risks.

## III. PROPOSED METHODOLOGY

Figure 2 shows the overall verification methodology based on LCSS algorithm. Thereafter, given an analog design described as a system of *ODEs*, the idea is to generate *SDEs* that describe the noise behavior. For process variation, technology vendors create a library of devices with different corners [10] that characterize the device in terms of power, speed, area, etc. This allows the designers to choose from a range of devices based on the application. In our proposed methodology, for  $0.18\mu\text{m}$  process, different circuit parameters are derived using Gaussian distribution with a known  $\pm 3\sigma$  deviation.

For environment constraints, this may include the amplitude of the noise, initial conditions of the circuit current and voltages. The *SDE* model, process variation, and the environment constraints are evaluated using Monte Carlo simulation in a MATLAB simulation environment. We then generate two

<sup>1</sup>Due to lack of space, the LCSS path finding algorithm is not provided.

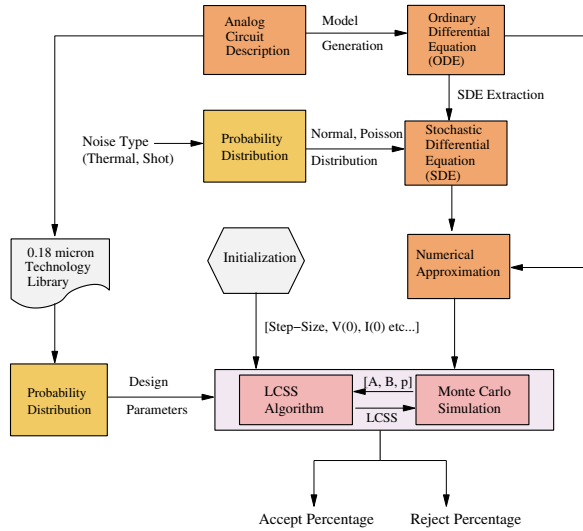


Fig. 2. Overview of the Verification Methodology

sets of sequences, one considered to be the sequence of an ideal circuit  $A$  (without noise and process variation) and the other sequence  $B$  of a non-ideal circuit (with noise and process variation). These two sequences ( $A$  and  $B$ ) along with the given tolerance level ( $p$ ) are evaluated using the LCSS algorithm as shown in Figure 2 in order to generate the LCSS. Based on the given confidence level, we can decide on either accepting or rejecting the circuit.

The basic idea behind the Monte-Carlo method is to generate multiple trajectories and sample them in order to calculate the desired statistics for a given confidence level. The circuit circuit state variables (current, voltage, etc.) are simulated for  $M$  Monte-Carlo trials to generate two sets of sequences. In general, there is no theory that governs the number of trials in Monte-Carlo simulation. However, a trade-off exists between the number of trials and the simulation run-times. The higher confidence can be gained by choosing a larger number of samples, but at the cost of run-times [9].

#### IV. APPLICATION - COLPITTS OSCILLATOR

The circuit diagram for a MOS transistor based Colpitts oscillator is shown in Figure 3. For the correct choice of component values, the circuit will oscillate due to the bias current and negative resistance of the passive tank. The frequency of oscillation is determined by  $L$ ,  $C_1$  and  $C_2$ . For simplicity, we assume the noise only from the passive elements.

##### A. Finding the Longest Closest Subsequence

The first step in finding the LCSS, is to generate two sets of sequences (ideal and non-ideal) for different cases as shown in Figure 4. For such simulation results, the property of interest is: “Whether or not for the given set of parameters, the inductor current is within a certain bound?”

Figures 4 (a) to (d) show the simulation results for the Colpitts oscillator under both ideal and non-ideal conditions. Figure 4 (a) shows the results in the absence of noise and process variation, which will be considered as an ideal output. From the Figures 4 (b) to (d), we note that, for the total

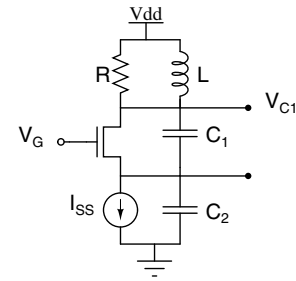


Fig. 3. Colpitts Oscillator

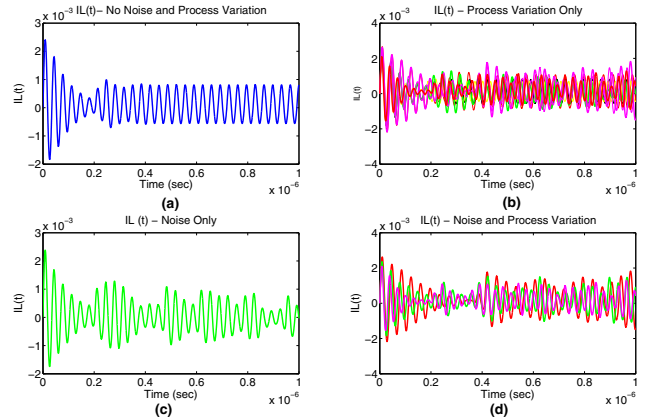


Fig. 4. Colpitt’s Simulation Results

simulation time of  $1.0 \times 10^{-6}$ , we come across some increase in amplitude for the circuit influenced by noise and process variation at certain simulation points. This is because, the variation in device parameter and additive noise has caused an increase in the inductive current thereby creating a non-uniform output. The question now is: “Do we have to reject those circuits entirely?” To assist the designers in making a better decision, we can determine for different tolerance levels, the LCSS for different conditions using the technique described in Section II. All together, we have one ideal and seven different (individual/combined effect of noise and process variations) Colpitts oscillator circuit implementations. For each of the seven different circuits, we compare the sequence with the ideal sequence in order to generate the LCSS (*slow, nominal and fast*) as summarized in the Table I.

TABLE I  
LONGEST CLOSEST SUBSEQUENCE COMPUTATION RESULTS.

Tolerance (%)	No. of LCCS (PV = Process Variation, m = 1000)						
	Noise	Process Variation			Noise & Process Variation		
		Slow	Nominal	Fast	Slow	Nominal	Fast
0.1	472	237	1000	2	235	440	2
0.5	585	341	1000	6	334	559	8
1.0	603	345	1000	12	338	573	16
5.0	801	366	1000	53	358	705	71
10.0	971	374	1000	100	385	976	129

The experiments were conducted for different tolerance levels and for a sequence length of  $m = 1000$  between the two outputs. In the presence of noise, for a given tolerance level, the additive Wiener process in the SDE model makes the inductor current to deviate from its specified value, thereby creating discrepancies between the ideal circuit and the noisy circuit as evident from the Table I (column 2). A tolerance level of 0.1% means that the output sequence of the non-ideal circuit is within  $\pm 0.1\%$  range of the ideal circuit output. It is also apparent from columns 3-5, that the analysis with parameter variation due to  $0.18\mu\text{m}$  shows little effect for the nominal process and adverse effect for the *fast* process corner. This is because  $\pm 3\sigma$  parameter variation is large enough to create a discrepancy on the inductor current for *worst* case process rather than *nominal* case. Hence, for those 1000 sequence length the acceptance is 100% in case of nominal process. In contrast, in columns 6-8 of Table I, it is evident that the effect of noise and process variation have led to minimum number of matches between the two sequences.

Though we have achieved a good insight on the matching sequence length, we may come across false oscillation due to noise and process variation. For instance, there may be a situation wherein, for the given set of parameter values, the Colpitts circuit can have no oscillation for the ideal case and false oscillation due to noise and process variation as shown in Figure 5. In that case, if the tolerance parameter is large, then the LCSS algorithm might consider a matching case. Hence, the LCSS derived in this case is considered to be *false positive*. For the Colpitts oscillator, such a false simulation can be determined using the confusion matrix (Table II).

For demonstration of the confusion matrix, we have taken four sample points from the simulation results and for  $i = 1 \dots 4$ , we define two values for  $X$  ( $X_{iH}$ ,  $X_{iL}$ ) and  $Y$  ( $Y_{iH}$ ,  $Y_{iL}$ ) representing the lower and upper limits of a non-ideal/ideal circuit sequence, respectively. For a given tolerance range (7% in this case), *true positive* (TP) represents the number of correct predictions that an LCSS is closely matching. From Figure 5, even though we can derive a LCSS between  $2.0 \times 10^{-6} \leq \text{Time} \leq 2.5 \times 10^{-6}$ , it is considered to be *false positive* (FP) due to its oscillation. *False positive* represents the number of incorrect predictions that an LCSS is closely matching. A *false negative* (FN) represents the number of correct predictions that are not closely matching.

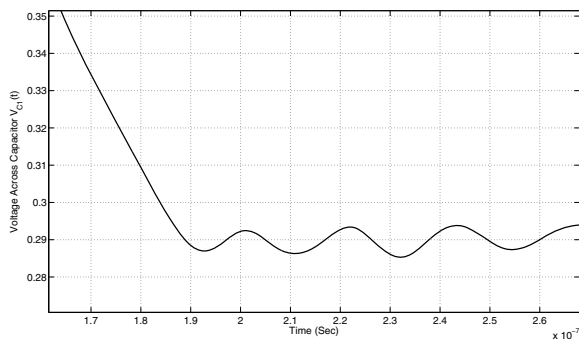


Fig. 5. False Oscillation for Colpitts Oscillator.

TABLE II  
CONFUSION MATRIX FOR THE COLPITTS OSCILLATOR CIRCUIT.

TP - True Positive, FP - False Positive, FN - False Negative								
Y (V)	X <sub>1</sub> (V)		X <sub>2</sub> (V)		X <sub>3</sub> (V)		X <sub>4</sub> (V)	
	X <sub>1H</sub>	X <sub>1L</sub>	X <sub>2H</sub>	X <sub>2L</sub>	X <sub>3H</sub>	X <sub>3L</sub>	X <sub>4H</sub>	X <sub>4L</sub>
Y <sub>1</sub>	TP	TP	FP	FP	FP	FP	FN	FN
Y <sub>2</sub>	TP	TP	FP	FP	FP	FP	FN	FN
Y <sub>3</sub>	TP	TP	FP	FP	FP	FP	FN	FN
Y <sub>4</sub>	TP	TP	FP	FP	FP	FP	FN	FN

In general, the combined Monte Carlo and LCSS analysis can be different for tolerance levels and the accuracy would be compromised if the tolerance level is too high or the number of trials being too low. Higher tolerance levels would increase the error margin and degrade the reliability.

## V. CONCLUSION

In this paper, we have presented a methodology based on pattern matching for noise and process variation in analog designs. Our approach is based on modeling the noise using SDEs, and then integrate the device variation due to  $0.18\mu\text{m}$  fabrication process for finding the LCSS and confusion matrix for a given tolerance level. We have demonstrated the methodology on a Colpitts oscillator circuit. The main advantage of the proposed methodology are robustness and the flexibility to account for wide range of variations, scalability, etc.

The proposed methodology is limited to finding LCSS for a fixed simulation step-size and we are working towards extending it to handle varying step-size as in the case of a design that has been simulated by two different simulators with different fixed/varying step-sizes. In addition, we would like to conduct the LCSS analysis to determine the acceptance or rejection using hypothesis testing [9].

## REFERENCES

- [1]  $0.18\mu\text{m}$  CMOS Fabrication Process. <http://www.tsmc.com>, 2008.
- [2] B. Ankele, W. Hölzl, and P. O'Leary. Enhanced MOS Parameter Extraction and SPICE Modeling for Mixed Signal Analogue and Digital Circuit Simulation. IEEE International Conference on Microelectronic Test Structures, pp. 133-137, 1989.
- [3] B. Oksendal. Stochastic Differential Equations. An Introduction with Applications. Springer, 2000.
- [4] E. Clarke, A. Donze and A. Legay. Statistical Model Checking of Mixed-Analog Circuits with an Application to a Third-Order Delta-Sigma Modulator. Haifa Verification Conference, LNCS 5394, pp. 149-163, Springer, 2008.
- [5] P. A. Gray, P. J. Hurst, S. H. Lewis and R. G. Meyer. Analysis and Design of Analog Integrated Circuits, Wiley, 2009.
- [6] P. Paper, M. J. Deen and O. Marinov. Noise in Advanced Electronic Devices and Circuits. AIP International Conference on Noise in Physical Systems and 1/f Fluctuations, pp. 3-12, 2005.
- [7] R. Narayanan, M. Zaki, and S. Tahar. Using Stochastic Differential Equation for Verification of Noise in Analog/RF Circuits. Journal of Electronic Testing: Theory and Applications, 26(1):97-109, Springer, 2010.
- [8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms, The MIT Press, 2001.
- [9] W. L. Martinez and A. R. Martinez. Computational Statistics Handbook with MATLAB. Chapman & Hall/CRC, 2002.
- [10] Y. Cheng. The Influence and Modeling of Process Variation and Device Mismatch on Analog/RF Circuit Design. IEEE International Conference on Devices, Circuits and Systems, pp. 1-8, 2002.