Using LCSS Algorithm for Circuit Level Verification of Analog Designs

Rajeev Narayanan, Alaeddine Daghar, Mohamed Zaki, and Sofiène Tahar
Dept. of ECE, Concordia University Montreal, Quebec, Canada
Email: {r_naraya, daghar, 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 process variation and mismatch in analog circuits. At circuit level, the effect of mismatch and process variation that results in offsets is analyzed by performing parametric and statistical techniques and then applying LCSS to estimate the probability of closest matching. The acceptance/rejection of a circuit is done using bounded hypothesis testing. The approach is illustrated on a Rambus ring oscillator circuit for a *90nm* fabrication process. Advantages of the proposed methods are robustness and flexibility to account for a wide range of variations.**

I. Introduction

Verification of analog designs is faced with immense challenges with the uncertainties due to unwanted deviation in a signal trace because of component mismatch and process variations. The effort to nullify the effect of *DC* offset [3] using an offset cancelation circuit has remained popular among analog designers [11]. On the other hand, *horizontal* offset common in oscillators, can be considered to be a complete drift in the time axis, which can be a *start-up* delay. Though, the offset analysis is done at circuit level, developing a methodology that could verify an analog circuit with offset condition has not yet been addressed. This paper addresses the above issues, by automatically ensuring the correctness of analog circuits in the presence of *offset* conditions using the concept of pattern matching.

Pattern matching techniques are commonly applied to the characterization and validation of high-speed analog and digital circuits. Quite often they are associated with the study of crosstalk, coupling, delays in the data transmission lines [5] during the post-layout and board-level signal integrity (SI) analysis. In the current state-of-the art, popular quantitative methods in the form of assertion/statistical approaches [9] can sometimes exhibit violations that may not be associated with real design failures and fall short to enumerate the method of failure for the circuit behavior appropriately. This issue can be addressed by extending the pattern matching concepts developed in SI analysis to the functional verification.

The Longest Common Subsequence (LCS) is a pattern matching algorithm that finds its applications in computational biology, chip layout design, and so on [12]. The underlying idea of this algorithm is to find the subsequence simulation trace between a set of analog signal traces and then use the combination of MonteCarlo and hypothesis testing to determine the probability of acceptance/rejection of those traces. 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. The extension to LCS in the form of the longest closest subsequence (LCSS) algorithm that has been presented in [8] cannot handle offset conditions which are addressed in this paper.

II. Proposed Methodology

Figure 1 shows the circuit-level simulation methodology that involves parametric and MonteCarlo simulations. First, we begin with an analog circuit description as a schematic entry that is simulated for a specified process and for a specific initial condition using a SPICE simulator [10]. Parametric analysis involves sweeping multiple parameters to help analyze the stability of a solution within the specified tolerance zone. On the other hand, the basic idea behind the MonteCarlo method is to sample the model of the true population of interest and then to determine the statistical outcome of the simulation. The sampling and calculation procedure is repeated for "M" trials.

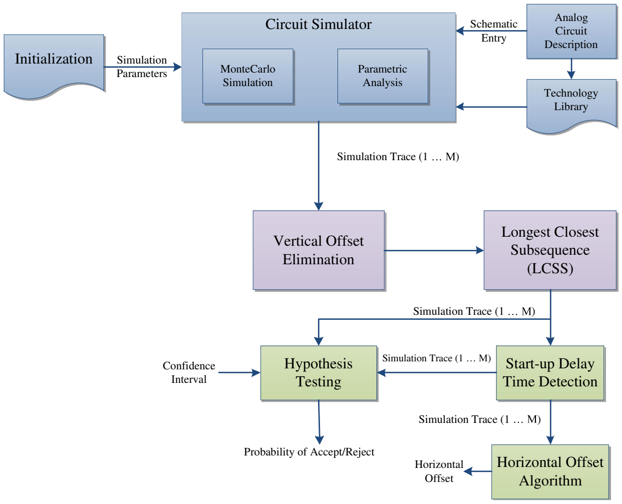

Fig. 1. Circuit Level Simulation

For analog circuits, the MonteCarlo technique is used to study the effect of random variations due to process and mismatch. Mismatch could be either "systematic", where values are fixed and known, or it could be "random", where the values are generated randomly and often unknown [10]. In general, a

978-1-4673-0859-5/12/$31.00 ©2012 IEEE

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 [7]. The question that has to be answered now is: "*how to decide on the sequences that have offset conditions?*"

As depicted in Figure 1, LCSS and hypothesis testing are combined to parametric analysis and process variation for the circuit level simulation traces that have offset conditions to determine the probability of acceptance/rejection.

Start-up Delay Time Detection: Start-up delay time can be considered as one kind of horizontal offset. When the LCSS algorithm is applied, if the non-ideal trace does not find a match from its first values, it is considered as a start-up delay time as shown in Figure 2.

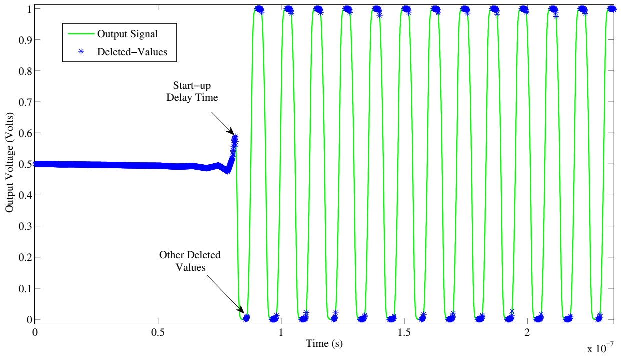

Fig. 2. Start-up Time Delay Detection

The implementation of the start-up delay time detection is given in Algorithm 1.

Algorithm 1 : Start-up Algorithm

```
Require: $deleted\_values$
 1: $d \leftarrow deleted\_values$
 2: $l \leftarrow length(d)$
 3: for $j \leftarrow$ 2 to $l$ do
 4:     $sp \leftarrow d(j) - d(j-1)$
 5: end for
 6: $spacing \leftarrow eliminate(sp < dist)$
 7: $ls \leftarrow length(spacement)$
 8: $MeanSpacing \leftarrow \sum(spacing)/ls$
 9: $SMS \leftarrow MeanSpacing * SecurityCoffiecient$
10: $ind \leftarrow 1$
11: while $d(ind+1) < d(ind) + SMS$ do
12:     $ind \leftarrow ind + 1$
13: end while
14: $index \leftarrow d(ind)$
15: $startup \leftarrow time(index)$
16: return $startup$
```

The algorithm calculates the distance between every two deleted points (lines 3-5). If this distance is greater than a threshold as defined by the user (line 6), then this value will be taken. Otherwise, it represents two successive deleted regions for which the arithmetic mean to estimate the spacing (line 8) distance is computed. The user also has to specify a security coefficient usually (< 1) to be multiplied by this spacing distance (line 9). In all cases, it is assumed that the start-up delay time occurs from the time 0s (first index), which is very natural (line 10). The spacing distance is then incremented (lines 11-13) to estimate the distance between two deleted points when it is not a start-up time.

Horizontal Offset: A horizontal offset consists of a shift between two output traces in the time domain. This shift in time is represented as a shift in the index on the two sets of sequences. As LCSS performs operations on a set by set basis rather than value by value, detecting or eliminating offsets will depend on the correlation between the ideal and non-ideal sequences. The implementation of horizontal offset is described in Algorithm 2. It can be calculated using the MATLAB built-in correlation function (*xcorr*). The correlation between two signals is maximal when they are aligned. The horizontal offset time (line 3) is thus measured by detecting the index of the maximum of correlation as described in line 2.

Algorithm 2 : Horizontal Offset Algorithm

```
Require: $X$, $Y$,
1: $[cc, lags] \leftarrow xcorr(X, Y)$
2: $offIndex \leftarrow max(cc)$
3: $offset \leftarrow time(offIndex)$
4: return $offset$
```

III. Application - Rambus Ring Oscillator Circuit

The efficiency of the proposed methodology is illustrated on a Rambus ring oscillator [6] circuit for a *90nm* fabrication process. This Rambus ring oscillator circuit presents a unique problem of lock-up. Unlike traditional ring oscillators that have an odd number of inverters, the Rambus ring oscillator consists of an even number of stages (say *"n"*), with a bridge between each stage as shown in Figure 3.

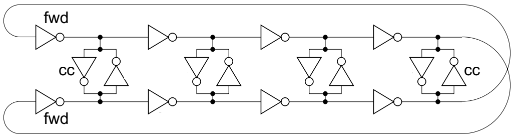

Fig. 3. Rambus Ring Oscillator Circuit.

If the "forward" inverters (labeled *fwd*), are much larger than the "cross-coupling" inverters (labeled *cc*), then the circuit acts like a ring of *2n* inverters and will not oscillate. The same problem occurs if the *cc* inverters are much larger than the *fwd* inverters. The oscillation also depends on the circuit initial condition. While designers can establish conditions that ensure a stable ("good") oscillation, offset can sometimes make the oscillation look "bad" or "ugly". The challenge for the verification engineers is to answer the question *"how to judge the quality of oscillation?"* and *"how can we detect the offsets automatically?"*

We can define the transistor sizing ratio by [4]:

$$r = \frac{size\ of\ cc}{size\ of\ fwd} \tag{1}$$

To better answer the question, the output of the circuit is analyzed by sweeping the transistors size (ratio r) and the initial condition (parametric analysis). Then, MonteCarlo simulation for 100 trials is performed to study the impact of *90nm* technology variation and mismatch on the output behavior of the oscillator. In both cases, LCSS is applied to the output sequence to determine the probability of matching and start-up delay time.

Parametric Analysis: Based on our simulation and results reported elsewhere [4], it is concluded that the oscillator is unstable (so it oscillates) if $r \in [0.52,\ 2.61]$ with the initial condition equal to the supply voltage for *90nm* technology. Though, the authors in [4] have demonstrated that the circuit will enter an oscillation stage from any initial condition, the challenge would be to verify the quality of oscillation in terms of offsets and start-up delay time. By doing so, the designers will have the leverage to trade-off between the desired frequency range and the speed with which the oscillator can start, meaning the delay time. This is needed as the ring oscillator could be a part of a larger analog and mixed signal (AMS) circuit such as PLL, where the start-up delay time can be related to the lock time.

For parametric analysis, the idea is to sweep *r* for each initial condition and then compare the output simulation trace with an ideal oscillator, which has the same frequency using the LCSS pattern matching algorithm. Then, we calculate the percentage of matching and the start-up delay time.

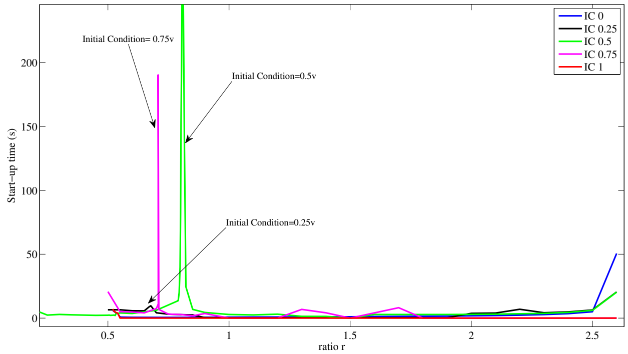

Fig. 4. Start-up Delay Time as Function of r

Finding the LCSS and Start-up Delay Time: Figure 4 shows the analysis results of the start up delay time as function of *r* for different initial condition values (*IC*). By sweeping the ratio *r*, the delay time changed remarkably, especially for *IC=0.5v* and *IC=0.75v*. The parametric analysis gives little information about the start-up time. This is true because at any given point, the components values are taken as fixed, which is not the case for analog circuits in reality. Hence, there is a need for a statistical method in the form of MonteCarlo simulation to study the impact of those variations on the output voltage.

MonteCarlo Simulation: For *90nm* technology, and based on the number of trials "M", there are many outputs for the same design. These outputs are the result of varying components values due to the process and mismatch and follow certain probability distribution. The experiment for M trials is conducted on a SUN UltraSPARC-III with 4GB memory.

Finding the LCSS: From the simulation data through statistical analysis, it can be determined that the percentage of matching fits different distributions for different values of *r* as summarized in Table II. For process variation, it is quite natural to find Normal and Lognormal distributions [1] as process variation was performed using a Normal distribution for the parameter's distribution. However, for certain values of *r*, the data fits Weibull distribution. The Weibull distribution is a general purpose distribution that can be used to represent normal, exponential and other distributions. For certain values of *r*, the distribution appears to be skewed from its mean and hence for such cases it is better to represent them as Weibull than as Normal or LogNormal distribution [2].

r	Maximum Frequency (MHz)	# of Percentage of Matching	Minimum Startup Delay Time (ns)
0.5	92	19 (72.4%)	2.0
0.75	86	35 (69%)	0.2
1.0	82	38 (67%)	0.1
1.25	76	20 (63.8%)	1.6
1.5	70	19 (61.8%)	40.0
1.75	60	22 (59%)	2.8
2.0	52	26 (56.6%)	3.2
2.25	41	16 (54.2%)	4.0
2.5	24	24 (47%)	6.0

TABLE I
PARAMETRIC ANALYSIS WITH INITIAL VALUE=0.5VOLTS

Start-up Delay Time Estimation: Figure 5 shows the statistical distribution of the MonteCarlo simulation results for estimating the start-up delay time for different *r* values. The variations are not only on the mean but also on the variance. This will allow us to have a better idea on how accurate our estimations are. Table I summarizes the MonteCarlo results for finding the LCSS and fastest start-up delay time for an initial condition of 0.5 volts. Therein, it can be noted that the optimal value for the frequency is 82 MHz with 38 out of 100 circuits having maximum percentage matching. When a sufficiently large number of trials is used in the MonteCarlo simulations, as per the central limit theorem [7], the estimation of the mean is fairly accurate.

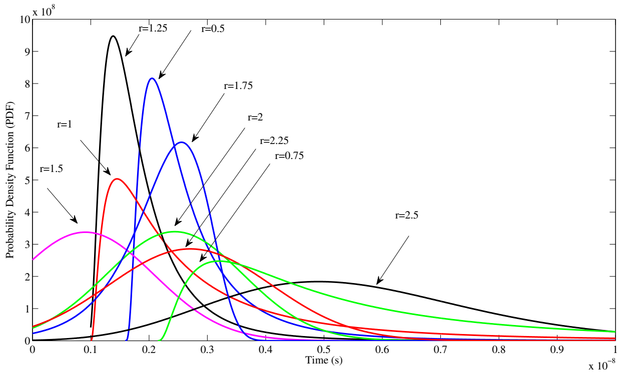

Fig. 5. Start-up Time as Function of r With MonteCarlo Simulation

Decision Based on Hypothesis Testing: Table II summarizes the experimental results for the Rambus ring oscillator circuit based on hypothesis testing. It can be seen that, the results for the frequency and percentage of matching are consistent with those results found during the parametric analysis,

r	Frequency (MHz)		Percentage of Matching (%)		Start-up Delay Time (ns)	
	Distribution	Acceptance region	Distribution	Acceptance region	Distribution	Acceptance region
0.5	LogNormal	[82.11 – 99.28]	Weibull	[71.13 – 73.61]	LogNormal	[1.82 – 4.65]
0.75	LogNormal	[78.28 – 94.15]	Weibull	[66.51 – 70.01]	LogNormal	[2.69 – 15.05]
1	LogNormal	[73.97 – 88.31]	Weibull	[64.11 – 67.41]	LogNormal	[1.24 – 8.11]
1.25	LogNormal	[68.93 – 81.78]	Normal	[62.84 – 64.63]	LogNormal	[1.16 – 3.37]
1.5	LogNormal	[63.61 – 74.68]	Weibull	[60.48 – 62.37]	Weibull	[0 – 2.85]
1.75	LogNormal	[56.67 – 65.95]	LogNormal	[58.38 – 59.71]	Weibull	[0.79 – 3.21]
2	LogNormal	[48.61 – 55.98]	LogNormal	[56.02 – 57.25]	Normal	[0.51 – 4.37]
2.25	LogNormal	[37.88 – 43.93]	LogNormal	[53.21 – 54.68]	Normal	[0.408 – 5]
2.5	LogNormal	[16.51 – 28.48]	LogNormal	[42.22 – 50.41]	Normal	[0.86 – 10.38]

TABLE II

HYPOTHESIS TESTING RESULTS

meaning that the mean of the frequency and percentage of matching decreases with increasing *r*. However, compared to the standard deviation associated with the frequency, the percentage of matching shows different standard deviations but still remains small for *r=2.25* as shown in Figure 6.

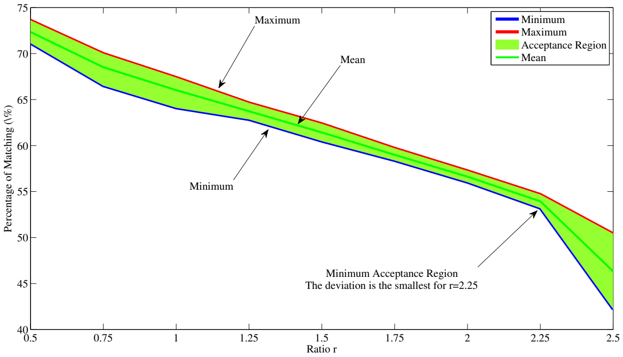

Fig. 6. Percentage of Matching Acceptance Region as Function of r

Figure 7 plots the mean with its corresponding acceptance region for the start-up delay time. The results shows the adverse influence of process variation and mismatch. The start-up time exhibits a larger acceptance region where $r \simeq 0.75$ which confirms the results found in the parametric analysis for the initial condition 0.5v. For $r \simeq 0.80847$ and for the initial condition 0.5v, the oscillator takes a huge time to start compared to other *r* values. In summary, the hypothesis testing results can be different for different confidence intervals and the accuracy would be compromised if the confidence level is too high or too low. Higher confidence level would increase the error margin and degrade the reliability; lower confidence level on the other hand would increase the rejection region and cause low accuracy. The confidence level of 100% is impossible to reach.

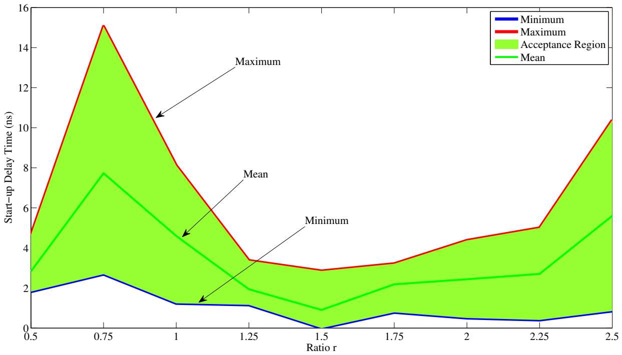

Fig. 7. Start-up Time Acceptance Region as Function of r

IV. CONCLUSION

This paper describes a methodology based on pattern matching to account for process variation and mismatch in analog circuits. The longest closest subsequence (LCSS) algorithm is used at circuit level for the qualitative analysis on the simulation traces that have *offset* conditions. The effect of mismatch and process variation at circuit level is studied by performing parametric and statistical analysis to estimate, in terms of percentage, the closest simulation trace that matches with the simulation trace of an ideal circuit. The efficiency of our approach is illustrated on a Rambus ring oscillator circuit for a *90nm* fabrication process.

Our future plan is to develop techniques that could handle *frequency* offset conditions and address the issue related to stability of analog circuits. Additionally, the algorithm has to be optimized for speed and memory utilization. For instance, this can be done by using threading techniques for the LCSS implementation.

REFERENCES

[1] 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.

[2] Easy Fit Online Help. `http://www.mathwave.com/help/easyfit/index.html`, 2012.

[3] P. A. Gray, P. J. Hurst, S. H. Lewis and R. G. Meyer. Analysis and Design of Analog Integrator Circuits, Wiley, 2009.

[4] M. Greenstreet, and S. Yang. Verifying Start-Up Conditions for a Ring Oscillator, ACM Great Lakes Symposium on VLSI, pp. 201-206, 2008.

[5] R. J. Haller. The Nuts and Bolts of Signal-Integrity Analysis. Electronics Design, Strategies, News, pp. 61-78, March 2000.

[6] K. D. Jones, J. Kim, and V. Konrad. Some "Real World" Problems in the Analog and Mixed Signal Domains. International Workshop on Designing Correct Circuits, pp. 51-59, 2008.

[7] W. L. Martinez and A. R. Martinez. Computational Statistics Handbook with MATLAB. Chapman & Hall/CRC, 2002.

[8] R. Narayanan, M. Zaki, and S. Tahar: Ensuring Correctness of Analog Circuits in the Presence of Noise and Process Variation Using Pattern Matching, IEEE/ACM Design Automation and Test in Europe, pp. 1188-1191, 2011.

[9] G. Al Sammane, M.H. Zaki, Z.J. Dong and S. Tahar. Towards Assertion Based Verification of Analog and Mixed Signal Designs Using PSL. Forum on Specification and Design Languages, pp. 293-298, 2007.

[10] Synopsys HSPICE User Guide: RF Analysis. `http://www.synopsys.com`, 2012.

[11] J. F. Witte, K .A. A. Makinwa and J. H. Huijsing: Dynamic Offset Compensated CMOS Amplifiers (Analog Circuits and Signal Processing), Springer, 2009.

[12] M. Yoshikawa and H. Terai. Constraint-Driven Floorplanning based on Genetic Algorithm. ACM International Conference on Computer Engineering and Applications, pp.147-151, 2007.