EE595 SSTA Phase I Report

Zhiyu Chen
zhiyuc@usc.edu

Hsu Cheng
hsucheng@usc.edu

Harmanpreet Singh Kalsi
hkalsi@usc.edu

***Abstract*—The concept of process variation has been around as old as semiconductor history. It is the naturally occurring variation in the attributes of transistors when ICs are fabricated. As the technology evolves over time, the amount of process variation becomes more pronounced at smaller process nodes. While static timing analysis has been one of the pervasive analysis method over the last 30 years, the decreasing predictability in semiconductor devices due to the process variation raised some questions over the ability of STA to effectively model the process variation. This lead to more extensive research on statistical static timing analysis (SSTA) which made a significant leap forward from the traditional STA methodology. In this paper, we review the concept of process variation and how SSTA has developed in order to model the complex process variation.**

This report will give a brief introduction to the Statistical Static Timing Analysis (SSTA). And will present the main challenges in terms of software and algorithms development. Some of the results of the implementation will be discussed.

I. Introduction

The primary motivation of this report is to introduce the concept of Statistical Static Timing Analysis and its importance in process variation in a more educational and presentable manner. Starting with the basics of the process variation, we will go over more thorough explanation and move on to the previous works done on the static timing analysis to discuss its limitation. After analyzing the constraints due to the decreasing feature sizes of integrated circuits, we will introduce SSTA and its applications on current and future integrated circuit manufacturing. This report will serve as an collection of essences on existing papers and articles on process variation and SSTA. Through this learning process of researching and writing the report, our goal is to become substantially knowledgeable on the subject by the end.

II. Literature Review

For the Literature review part, two main papers [1] [2] have been go through in detail. In paper [2], the author first discussed the importance of timing closure and Static Timing Analysis (STA) for the VLSI industry. Than, the topic extend to advanced process nodes, where the author explains why the traditional STA could not satisfied the need. After that, a few solutions, including SSTA have been proposed, and their corresponding pros and cons have been mentioned. Overall, this paper gives us a general introduction to the idea behind timing analysis and addressed the challenges in performing it.

Paper [1], gives a in depth view on challenges and corresponding solutions in doing timing closure on advanced process nodes. By presenting different sources of variations, the author first states the insufficiency of traditional STA in predicting the timing issue of model process technology. Then, multiple solutions, proposed by previous scholars have been discussed in detail, where the most appealing one is Probabilistic Analysis Methods. Unlike other methods which are sample-based, this method is divided into path-based and block-based. The paper paid more attention on the latter one. There are several blocked-based methods. The easiest one is Distribution Propagation Approaches (Gate-Delay Space) which assumed that both gate delays and latest arrival-time distributions are independent normal RVs. Based on this assumption, the sum and maximum of arrival-time RVs are computed using analytical results for the normal RVs. Finding upper and lower bounds, the correlation between two arrival times that are substantially shifted can be ignored without incurring significant error. The second method is Dependence Propagation Approaches (Parameter Space) which account for spatial correlation of the underlying physical device parameters. The basic difference between the two cases is that the correlation among arrival times for the second method now originates from the correlation of the device parameters. The Dependence Propagation Approaches can further model the spatial correlation as a grid or quadtree mode and a linear function of independent RVs. In the second model, the author further extended the linear model into quadratic one which seems to be more complicated. However, a Taylor-series expansion-based polynomial representation of gate delays and arrival times is able to effectively capture the nonlinear dependences. This paper points out problems that SSTA encounters and several possible solutions that may be the foundation of implementation of our project. It is a quite hard paper, but it is a good start.

Based on this two papers, other related ones have also been went through including Charles's book on probabilities [3], in which The basic idea about the Sum of two discrete random variable distributions is covered in this book.

Suppose X and Y are two independent discrete random variables with distribution functions $m_1(x)$ and $m_2(x)$. Let Z = X + Y. We would like to determine the distribution function $m_3(x)$ of Z. To do this, it is enough to determine the probability that Z takes on the value z, where z is an arbitrary integer. Suppose that X = k, where k is some integer. Then Z = z if and only if Y = z - k. So the event Z = z is the union of the pairwise disjoint events (X=k)U(Y=z-k)

where k runs over the integers. Since these events are

pairwise disjoint, we have

$$P\left(Z=z\right)=\sum_{k=-\infty}^{\infty}P\left(X=k\right).P(Y=z-k)$$

III. Experiment and implementation

A. Goal and motivation

In SSTA, all delays are modeled in probability distributions. For finding a delay in a certain point of the circuit, operations such as summation and maximization need to be done. In short, the delay at a given node of the circuit should equal to the maximum of sum of all the previous paths delay plus the delay of the node. However, this 'max of sum' operation seems to be identical to the 'sum of max' operation, in which the maximum of all previous delay is added to the delay of this node. So, in this experiment, our goal is to make a python implementation to check if these two cases are identical in distribution calculations.

B. Math for summation behind maximization for distributions

1) Summation: The conceptual proof for why the sum of 2 probability distribution models is doing the convolution can be displayed and observed easily from the equation of convolution.

$$(f*g)(t)=\int_{-\infty}^{\infty}f(\tau)g(t-\tau)d\tau \tag{1}$$

For the value t, there are several combinations for 2 sets of numbers to add up to be equal to it. The meaning of the convolution is that every time we move f a little bit (dτ) forward and multiply the probability of appearance of τ and (t-τ) and sum them up.

2) Maximization: To find the maximum between two distribution models, let's first assume two random variables X and Y. Then if Z is defined to be the maximum between X and Y we could write the equation as:

$$Z=max(X,Y)$$

Which could further be separated into two cases:

$$X\geq Y\ then\ Z=X->A$$

$$X\leq Y\ then\ Z=Y->\bar{A}$$

Therefore, the distribution function (CDF) of Z at point t can be expressed as:

$$F_z(t)\ =\ P(X\leq t)=P(Z\leq t,AU\bar{A})$$

Since A & $\bar{A}$, are two disjoint events, with some manipulations, we can get:

$$F_z(t)\ =P(X\leq t,X\geq Y)+P(Y\leq t,X<Y) \tag{2}$$

Then, if we look at this equation from a graphic point of view, Fig.1:

We can see that this equation, Eqn.2 also equals to:

$$F_z(t)=P(X\leq t,Y\leq t)$$

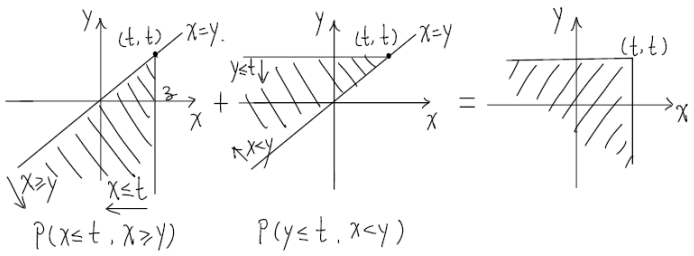

Fig. 1. Eqn.1 from a graphic point of view

Knowing X and Y are two independent events, we get:

$$F_z(t)=F_{X,Y}(t,t)$$

Base on the fact that the density function (PDF) is the derivative of distribution function (CDF), if we do the derivation on both sides by t, we get the density function of Z equals to:

$$f_z(t)=d/dt(F_{X,Y}(t,t))=f_x(t)F_y(t)+f_y(t)F_x(t) \tag{3}$$

If we take a further look at this function, Eqn.3, we find that it is saying the possibility of the maximum value at point t equal to the possibility of that point in X, times the total possibility of the value less than t in Y. Plus the possibility of that point in Y, times the total possibility of the value less than t in X.

For the implementation part of the max function, Zhiyu and Harman have come up with two different ways. The following part of this report will present both methods.

C. Implementation and results

1) Summation: To implement the SUM of two distribution model, ideally we could use the convolution to acquire the PDF. However, in reality we only have 2 sets of discrete data which save value and probability. We do not have any information of the sets of "t", so that we cannot directly use the convolution to acquire the PDF of sum. Instead, we compared every value in one set with the other and record the every possible value of "t", and simultaneously saved the probability of that combination of "t". After going through all combinations, we can acquire the distribution model of sum. To short, we used the Brute force method.

The result of our sum operation in python is shown in Fig.2

2) Maximization: For the implementation part of the max function, Zhiyu and Harman have come up with two different ways. The following part of this report will present both methods.

Max implementation by Zhiyu:

To implement the Max operation, two set of random variables, X and Y are first generated through 'random.normal()' function from numpy library in python. Normal distribution is chosen here because the results of Max operation on normal distribution is shown in the keynote paper by David Blaauw, we thus could have a reference to check the correctness of our result.

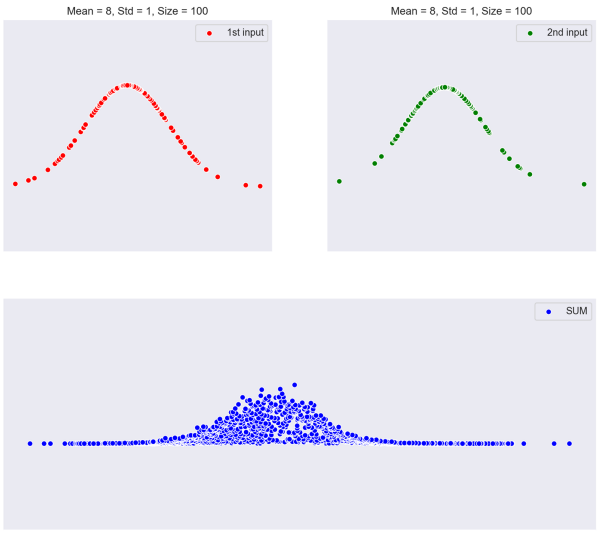

Fig. 2. Summation of two identical normal distributions

In addition, since the data we generated from the 'random.normal()' function are discrete values, we also made some observations and assumptions in finding the maximum:

- The range of Z(max(X,Y)) should be greater than the maximum between the lower bound of X and Y and should be smaller than the maximum of the upper bound of X and Y.
- If we could not find the exact same t from X in Y, instead of taking the PDF at t and CDF at t in Y as zero, we choose the PDF and CDF value of t-1, which is the immediate smaller value of t in random variable Y. We choose to make this assumption because if we have enough discrete data points, the PDF and CDF values are hardly to have large jumps, provides us with an acceptable approximation. Also, since this is done on sorted data, the time complexity for us to do one search for a given value of t is just O(n).

By having these observations and assumptions, the results we get are shown below (Red line is the max):

Since the graphical result seems to match with the result the author provides in the paper, we can say that this implementation of max function is valid for normal distributions.

Fig.6 shows the results from the article: [1]

Max implementation by Harman:

For both the data sets (x,y) find the approximate value of the pdf(x) and cdf(x) using the concept of interpolation. Define the new set of points with predetermined points. Use interpolation to find the value of pdf(x) and cdf(x) for each data point in the new set. This creates the uniform set of points for f_x (t),F_y

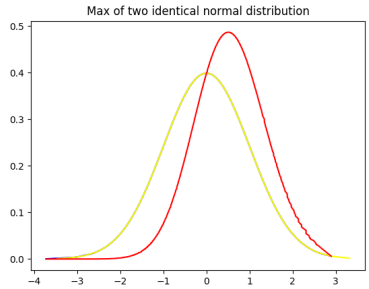

Fig. 3. Max of two identical normal distribution

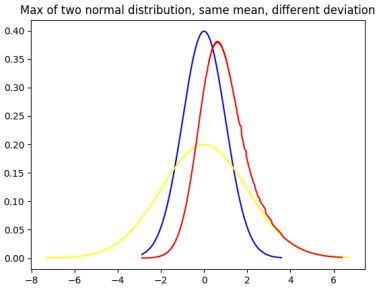

Fig. 4. Max of two normal distribution, same mean, different deviation

(t),f_y (t),F_x (t) and it is easy to get the max(x,y).

$$max(x,y) = \sum_{t=to}^{tn} (f_x(t)*F_y(t)+f_y(t)*F_x(t))$$

Assume,

$$x=\{x_1, x_2, x_3, ...x_n\}$$

$$y=\{y_1, y_2, y_3, ...y_n\}$$

Where x,y are sets of normally distributed random points.

Due to the random nature of these points it is difficult to find identical points in both x,y sets.So, in the proposed solution we create a new set of points z, such that

$$z=\{z_0\ ,z_0+\Delta,z_0+2\Delta,z_0+3\Delta, ..., z_0+(n-1)\Delta\}$$

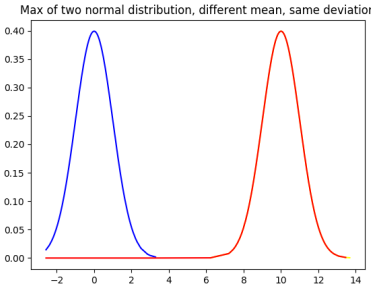

Fig. 5. Max of two normal distribution, different mean, same deviation

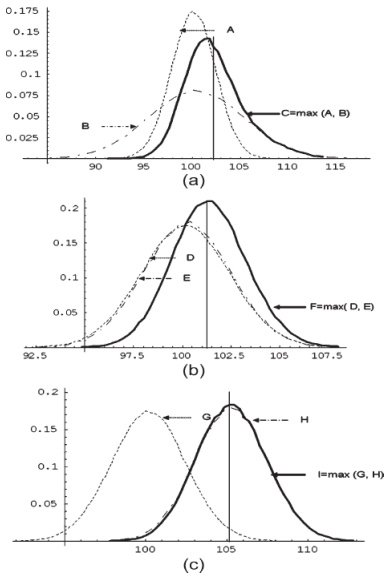

Fig. 6. Max operation shown by referenced paper

Where is a step size that can be chosen as per the precision required.

So, for the cases where $z_i < x_1$, assign $f_x(z_i) = 0$, $F_x(z_i) = 0$ where $z_i > x_n$, assign $f_x(z_i) = 0$, $F_x(z_i) = 1$
where $x_j < z_i < x_{j+1}$, assign

$$f_x(z_i) = f_x(x_j) + \frac{f_x(x_{j+1}) - f_x(x_j)}{x_{j+1} - x_j} * (z_i - x_j)$$

,

$$F_x(z_i) = F_x(x_j) + \frac{F_x(x_{j+1}) - F_x(x_j)}{x_{j+1} - x_j} * (z_i - x_j)$$

This way we get some finite value of $f_x(t), F_y(t), f_y(t), F_x(t)$ for all the points in z and the max operation simplifies.

Disadvantage: If the data points in x and y sets are widely distributed then to cover all the points, we need to incorporate a greater number of points in the z set. So, the computations also increase linearly, and it can take a lot of time to compute the results with this approach.

3) Sum of Max & Max of Sum comparision:: The result of the comparison between sum of max and max of sum are shown in Fig.7

As one can see, the result of sum of max and max of sum are different.While for further proving the difference, the T-statistics [4] has been calculated based on the mean and standard deviation of the two outcomes. By calculation, it shows this two distributions have a T value of 0.505, which means there are differences between them.

IV. Sampling Points Issue:

In the real world, we can acquire plenty of data for physical variables, such as threshold voltage, gate length and some physical parameters. But It is impractical and time-wasting

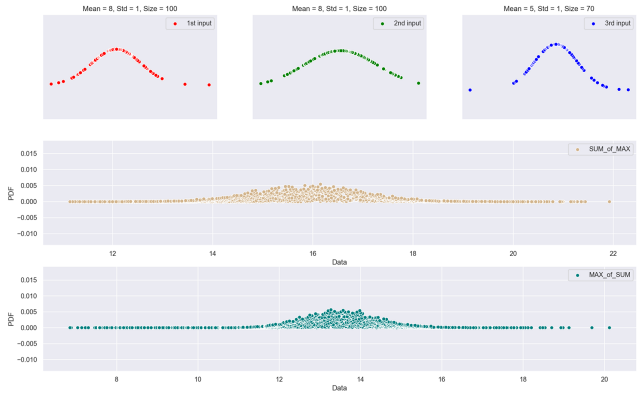

Fig. 7. Comparison between sum of max and max of sum of two normal distributions

to analyze and take all these data values into consideration. Therefore, how much portion of data should we take into consideration so that we can not only save time but also get the undistorted result for the further use becomes a critical issue. After taking some reviews and querying into some statistics references on "Sampling Distributions". The general way to do sampling is that we take N samples of the whole population and then we will find the sample means from these samples. We can finally get the so called "sampling distributions". When N becomes bigger, the sampling distributions will become close to the normal distribution. So, there is a rough rule of thumb in the field of Statistics. In order to prevent the result from being heavily distorted or skewed, Let's say if today the Population(P) that we care about and the Number of sizes of the sampling points that we are going to take are known, the relationship between of N and P is as follow:

NP >= 10, (Prevent Sampling Result from skewed left)

N(1-P) >= 10, (Prevent Sampling Result from skewed right)

Let's take an example:

If the fabrication manufacturer says that there are 2% of Vth of MOSFETs are at the 2nd standard deviation, which is in the region that will malfunction the chip from my company and there is a bunch of data for us to do the testing. So, we can select the sampling points using the criteria above

N >=10/0.02 = 500, (Prevent Sampling Result from skewed left)

N >= 10/0.98 ∼11, (Prevent Sampling Result from skewed right)

So, we should select at least 500 samples. Indeed, "Sampling distribution" seems to be a good way to decrease the data-analyzing time that SSTA needs. It allows us to use less data to acquire the PDF and CDF which are the basic information for analysis of SSTA. In the general situation of SSTA, most delays caused by plenty of factors such as process variations and environmental uncertainties are considered independent and normal-distributed, so that this "sampling distributions" works. However, in the real world,

we cannot promise that this rough rule will work well since the correlation of each factor exists and affects delay a lot. Maybe, we should estimate the number of sampling size more strictly and what kind of data from the foundry we can acquire is abstract and very limited, most information we acquire is from the higher level where it is defined more ideal than reality. The numerical analysis of this sampling size is not necessary, but we still can acquire some estimation from the practical experiment.

V. Project timeline

The tentative timeline for Phase II and III are:

Phase I: Discuss the viability of implementing sum and max operation in frequency domain using Fourier transform to accelerate the computation

Phase II: Implement the possible improvement we proposed in Phase II

VI. Work distribution:

Hsu-Cheng:

1) Designed the delay timing graph,
2) Complete the SUM operation
3) Figured out the empirical distribution model (which can be used in the condition when input is no more normal distribution)
4) Did the research on sampling points issue

Harman Preet Singh Kalsi:

1) Read the survey paper on Statistical Timing Analysis.
2) Implementation of max operation using interpolation.
3) Implementation of sum of max operation.
4) Working on the problem of max of sum.

Zhiyu Chen:

1) Math for sum and maximum operation
2) Implementation for max operation
3) Implementation for Max of sum and Sum of max comparison

Note: for Harman and Zhiyu, there are some overlapping part for the work. However, they both work differently. Because it is hard for verifying the correctness of sum of max operation, so if two different ways of implementations are done, we could have one more copy of reference to check with.

References

[1] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 4, 4 2008, pp. 589–607.

[2] S. Saurabh, H. Shah, and S. Singh, "Timing Closure Problem: Review of Challenges at Advanced Process Nodes and Solutions," pp. 580–593, 11 2019.

[3] "Grinstead and Snell's Introduction to Probability," Tech. Rep., 2006.

[4] "T-test using Python and Numpy - Towards Data Science." [Online]. Available: https://towardsdatascience.com/inferential-statistics-series-t-test-using-numpy-2718f8f9bf2f