EE201C final project report

Endi Xu, 005030030
Lu Shijun, 905035448

Introduction

In this project, we are supposed to apply stochastic modeling to estimate distribution of propagation delay in a given circuit. The circuit schema is show in figure 1 and the propagation delay is the delay between input side and output side.

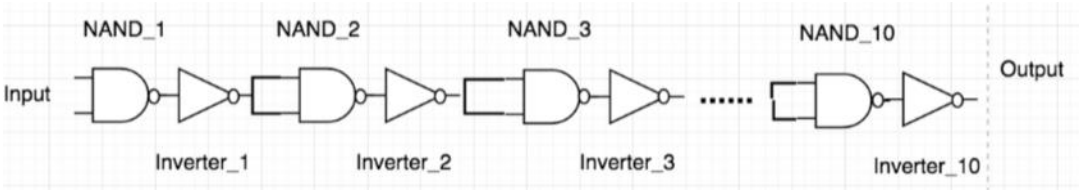

Figure 1 Circuit Schema

As shown in Figure 1, the circuit is made of 10 NAND gates and 10 inverters in cascade (30 NMOS and 30 PMOS in total). For each PMOS and NMOS, there are 6 random variables, where the distributions follow the Gaussian distribution. The means and standard variances of these variables are shown in table 1. The goal is to estimate the probability of circuit delay larger than 1.395e-10s.

	toxe	xl	xw	vth	u0	voff
mean(pmos)	2.700E-09	5.100E-09	1.800E-08	-3.96E-01	8.807E-03	-1.5E-01
std(pmos)	3.376E-20	4.277E-21	5.687E-20	1.150E-02	4.196E-05	1.797E-03
mean(nmos)	2.370E-09	5.800E-09	1.700E-08	3.29E-01	2.605E-02	-1.54E-01
std(nmos)	3.602E-22	4.681E-20	1.156E-19	1.094E-02	5.942E-06	1.367E-02

Table 1 random variables for PMOS and NMOS

Proposed Method

Our method used to estimate the failure probability is a combination of high dimensions important sampling (HDIS) [1] and hyperspherical clustering and sampling (HSCS) method [2]. Besides, a quick dimension reduction method is used first to simplify the problem. The details will be introduced in the following parts.

Step 1: Dimension Reduction

Based on the mean and sigma in table 1, the first 3 variables can be ignored since the sigma is much smaller than the mean. Then, we use control variates method to check the rest 3 variables. When one variable is tested, we set the sigma of the other 2 variables to 0. Then, run a couple of Monte Carlo simulation (about 1000) to check whether it has a big impact on the time delay.

The result shows that Vth is the dominate variable. The other variables have less influences on the failure rate. So, the original 360-dimension problem can be reduced to a 2-dimension problem (PMOS Vth and NMOS Vth).

Step 2: Pre-sampling

Because it is hard to find the final failure probability directly by Monte Carlo method. Alternatively, we

set a threshold delay time at 1.35e-10. By running Monte Carlo simulation with this new delay time, the simulation can be converged within 10k cases. The failure rate is about 0.057%. All the failure cases and pass cases need to be collected in this step for further analysis. The failure probability also needs to be recorded in this step.

Step 3: Remove useless points

Based on the data collected in pre-sampling step, we can plot the PMOS Vth mean and the NMOS Vth mean for each case on a 2D- dimension plot. which is shown in figure 2. Blue points show failure cases and red points show the pass cases. From figure 1, we can't tell which region on the plot is the failure region. So, a point reduce algorithm is designed to remove the failure points which can't show their differences from pass points. The proposed algorithm is the following:

```
For each failure cases
    For each pass cases
        Find the Euclidean distance between a failure point and its closest pass point
    end
    For each failure cases
        Find the Euclidean distance between a failure case and its closest failure point
    end
end
Remove failure points that is too close to pass points and failure points that is too fat to other failure points
```

Figure 1 Point Remove Algorithm

The reason to remove a failure point is very close to a pass point is because the failure point can't show the difference from pass points. It has a high probability located in pass region. Also, the reason to remove a failure point is far from other failure points is because the failure point is a single point in its region, which is not general. The figure 3 shows the new 2-D plot for PMOS Vth and NMOS Vth is generated. In figure 3, the difference between failure cases and pass cases is clearly shown. The failure region can be found.

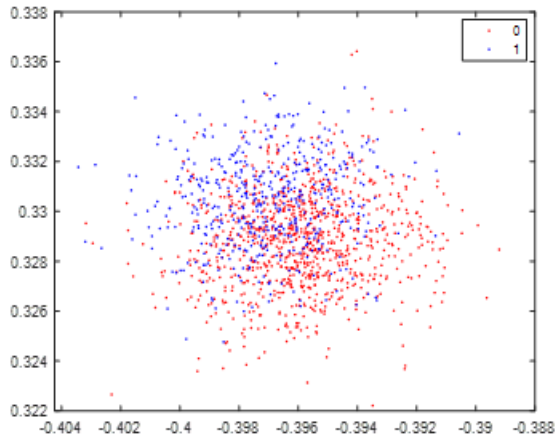

Figure 2 Original 2-D plot for PMOS Vth vs. NMOS Vth

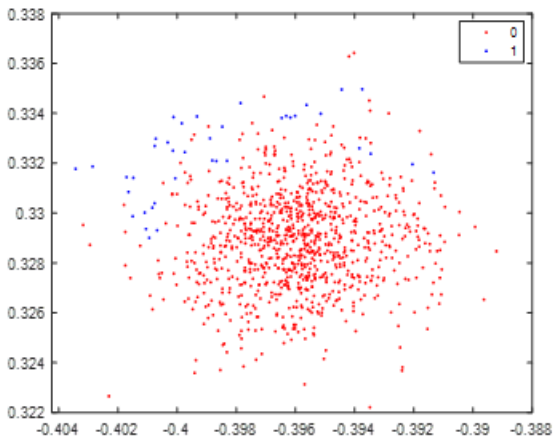

Figure 3 New 2-D plot for PMOS Vth vs. NMOS Vth

Step 3: Find Cosine Centroid and min-norm Point

Figure 3 shows failure region. However, these points are far from the final failure region ($t_d \geq$ 1.395e-10). So, K-mean algorithm is applied to find the final failure region. Figure 4 shows the failure points after using K-mean algorithm.

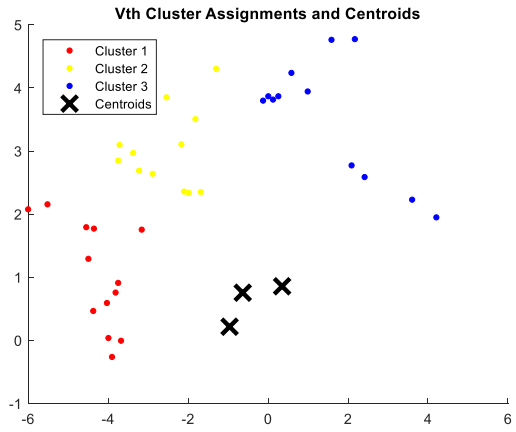

Figure 4 Failure points after clustering

The K-mean algorithm divides the failure points to 3 clusters. The 'X' sign is the cosine centroid of each cluster. Because the cosine centroids show the directions of failure region, the failure region will be reached in these directions. The min-norm point can be found by using the following algorithm. [2] The min-norm point is a 2-D vector (x, y). x refers to the NMOS Vth and y refers to the PMOS Vth.

Algorithm 2 Locate min-norm points for each cluster with bisection

Input: Minimal radius of existing failure samples, R
Output: Radius of min-norm point: R_{min}

$R_{max} = R$;
$R_{min} = 0$;
repeat
  $R = (R_{max} + R_{min})/2$;
  simulate a small set of samples at Radius $= R$ in current cluster;
  if any failed sample captured **then**
    $R_{max} = R$;
  else
    $R_{min} = R$;
  end if
until $R_{max} - R_{min} < R_{threshold}$
Return R;

Figure 5: Locate min-norm points

Step 4: Important Sampling for each cluster

After the final failure points is found, for each cluster, the mean of Vth is shifted to the min-norm point. The sigma of Vth is decided by the $max(abs(fPoint - Cmean), original\ sigma)$. Cmean refers to the Euclidean centroid of each cluster and fpoint refers to the min-norm point and Euclidean centroid of each cluster. Then, Monte Carlo simulation can be used for the important sampling to get $P_{IS} = \frac{P_{>1.395e-10}}{P_{>1.35e-10}}$ for each cluster. [1]

Step 5: Calculate the final probability

The final failure probability can be calculated by the following equation:

$$P_{final} = \sum_{i=1}^{3} P_{presampling} * \left(\frac{failure\ samples\ in\ cluster\ i}{total\ failure\ cases\ in\ presample}\right) * P_{IS_i}$$

The $P_{presampling}$ is the probability that the delay time is larger or equal than 1.35e-10, which is obtained from the pre-sampling step. The $\frac{failure\ samples\ in\ cluster\ i}{total\ failure\ cases\ in\ presample}$ is the ratio of selected failure points in each cluster and total failure cases. The P_{IS_i} is the important sampling result calculated for each cluster.

Experiment Result

Table 2 shows the general result.

Total samples needed	Failure Probability range
30k to 40k	**4e-6 to 7.5e-6**

Table 2: General Result

20K samples is enough to get the failure probability to 10^{-6} level. 30k to 40k samples can improve the accuracy. The final failure probability is 4e-6 to 7.5e-6. The probability range is decided by the point removing step and the number of cases generated in pre-sampling step.

The following is the result generated from one simulation:

Procedure	Number of Sampling	Number of failure samples	Failure Probability
pre-sampling	8584	494	5.754893e-02
Final failure points finding	900	N/A	N/A
Cluster1 sampling	2041	479	2.346889e-01
Cluster2 sampling	11355	269	2.369000e-02
Cluster3 sampling	7371	493	6.688373e-02
Final Result	**30251**	**N/A**	**6.2454e-06**

Table 2 Result from one simulation

The comparison of our method with Monte Carlo and HDIS method is shown in table 4.

	Monte Carlo	Original HDIS	This method
# simulation run	~3.2e8 (10667X)	~278K(9X)	~30K
Failure probability	5.3e-6	1.4226e-6	6.2454e-06

Table 3 Comparison for different methods

This method achieves 10667X speedup over Monte Carlo with a high accuracy. The tail distribution is shown in figure 6. The tail distribution is based on the result and fit is in to normal distribution.

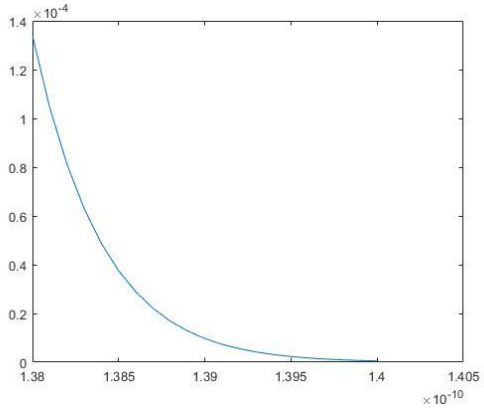

Figure 6 Tail Distribution

Further Improvement

This method is simple and fast. However, it is not a general method. It can only apply on this specific circuit. The problem can be reduced from 360-dimension to 2-dimension is because the PMOS and NMOS used in this circuit are the same and other variables except Vth have little impact on the time delay. In real circuit, this situation rarely happens. So, we should further improve this method to make it no matter works in high sigma but also in high dimension.

Reference

[1] W. Wu, F. Gong, G. Chen, and L. He, “A fast and provably bounded failure analysis of memory circuits in high dimensions,” in 19th ASP-DAC, 2014, pp. 424–429.

[2] Wei Wu, Srinivas Bodapati, and Lei He, "Hyperspherical Clustering and Sampling for Rare Event Analysis with Multiple Failure Region Coverage", International Symposium on Physical Design (ISPD), 2016.