Statistical Static Timing Analysis in the UCLA Timer

XinHeng Huang, John Lee, Puneet Gupta
University of California, Los Angeles, NanoCAD Lab

Introduction

Over the last thirty years, the deterministic static timing analysis has been sufficient for digital circuit design. However, in recent years the increased variations in digital circuit, such as perturbation in the fabrication process (Process Variations) and changes in the operating environment of the circuit (Environmental Variations), have introduced difficulties that cannot be handled well by deterministic static timing analysis. As a consequence, deterministic static timing analysis is insufficient now that parametric variations are growing. Statistical static timing analysis (SSTA) is the solution to account for both global and independent variations in digital circuit timing analysis.

Statistical Static Timing Analysis Method

Our statistical Static Timing Analysis (SSTA) follows the paper "First-Order Incremental Block-Based Statistical Timing Analysis" [1] that using Canonical Delay Model to represent nominal value, global correlations and independent randomness.

Canonical Delay Model:

$$A = a_0 + \sum_{i=1}^{n} a_i \Delta X_i + a_{n+1} \Delta R_a$$

$$B = b_0 + \sum_{i=1}^{n} b_i \Delta X_i + b_{n+1} \Delta R_b$$

Figure 1: The formula above describes the Canonical Delay Model. The a_0 represents the nominal mean, a_i represents the global sensitivities and a_{n+1} represents the independent sensitivity. The idea is the same for expression B.

Using the Canonical Delay Model to perform statistical "addition" function is trivial.

1) If the number of global sensitivities are the same for expression A and B . Then a_i will add with b_i .
2) If the number of global sensitivities are not the same. The missing number of global sensitivities from either expression A or B will be treated as 0 and then number of global sensitivities will match and add up correspondingly.

The independent sensitivities from expression A and B are not correlated. The statistical "add" operator will compute the effective variance by adding variance of independent sensitivities from expression A and B together and then output the result from the square root of the sum as the new independent sensitivity.

However, the statistical "maximum" function is a little bit more complicated because it will calculate the probability of distribution of max(A,B)

First of all, standard deviation θ of distribution A and B is computed:

$$\theta = \sqrt{\sigma_A^2 + \sigma_B^2 - 2\rho\sigma_A\sigma_B}$$

σ_A is calculated by the square root of the sum of variances of global sensitivity and independent sensitivity. The idea is the same for σ_B

The probability of expression A larger than B is T_A:

$$T_A = \emptyset(\frac{a_0 - b_0}{\theta})$$

a_0 and b_0 are the nominal mean

1) If the difference between a_0 and b_0 is larger than 5 times the standard deviation, the distribution with larger nominal mean will be returned because that distribution will dominate.
2) If the difference between a_0 and b_0 is less than 5 times the standard deviation, the mean will be calculated based on the following equation.

$$a_0 T_A + b_0(1 - T_A) + \theta * \emptyset(\frac{a_0 - b_0}{\theta})$$

Every individual global sensitivity (g_i) will be calculated by the following expression:

$$g_i = T_A a_i + (1 - T_A) b_i$$

The total variance will be calculated by the following expression:

$$(\sigma_A^2 + a_0^2) T_A + (\sigma_B^2 + b_0^2)(1 - T_A)$$
$$+(a_0 + b_0) * \theta * \emptyset\left(\frac{a_0 - b_0}{\theta}\right)$$
$$-\{E[\max(A, B)]\}^2$$

In order to find the independent sensitivity for max(A,B), the total variance calculated above will be used to subtract the sum of variances of global sensitivities, which will result the variances of independent sensitivities. Finally, the independent sensitivity for max(A,B) will just be the square root of the variances of independent sensitivities.

The statistical "minimum" function will take the negative of A and B as the input to the statistical "maximum" function. Then the negative of the output of the max function will be the return value of the statistical "minimum" function.

Implementation Sample

```
 // Calculating the standard deviation of A and B
double Z=(a.mean-b.mean)/std;
// Calculating the mean for max(A,B)
temp.mean=a.mean*phi(Z)+b.mean*(1-phi(Z))
          +std*Gauss_pdf(Z);
//Calculating the total variances
double
Var=(pow(a.sigma(),2)+pow(a.mean,2))*phi(Z)
  +(pow(b.sigma(),2)+pow(b.mean,2))*(1-phi(Z))
  +(a.mean+b.mean)*std*Gauss_pdf(Z)
  -pow(temp.mean,2);

if(b.global_sensitivity.size()<a.global_sensit
ivity.size()){
     for(int i=0;
     i<(a.global_sensitivity.size()-b.global_s
     ensitivity.size());i++)
          b.global_sensitivity.push_back(0);
}else{
     for(int i=0;
     i<(b.global_sensitivity.size()-a.global_s
     ensitivity.size());i++)
          a.global_sensitivity.push_back(0);
}
double sum=0;
//Calculating global sensitivities
for(int i=0;
i<max(a.global_sensitivity.size(),b.global_sen
sitivity.size());i++){
 temp.global_sensitivity.push_back(
    global_sensitivity[i]*phi(Z)
    +(1-phi(Z))*b.global_sensitivity[i]);
     sum+=pow(temp.global_sensitivity[i],2);
}
//Calculating the independent sensitivities
temp.indep_sensitivity=sqrt(Var-sum);
```

UCLA Statistical Timer Feature

UCLA Statistical Timer has two options, statistical and deterministic.

- If an input sensitivity file is specified, the UCLA Statistical Timer will calculate the Statistical quantities

- If no input sensitivity file is specified, the UCLA Statistical Timer will set the global sensitivities as well as the independent sensitivity to be 0 by default and calculate the deterministic quantities

For inputting the sensitivity file, a sensitivity file needs to be written, which will follow the following syntax:

cellName [space] indep = Number[space] global= Number,Number,Number

or

Instance_Name [space] indep = Number[space] global= Number,Number,Number

The instance_name has higher priority than the cell name. That is if an inverter (cell Name: INV, Instance_name: i_1) have two definitions of sensitivities, the one using the instance name syntax will overwrite its cell name's sensitivities.
The specification flag for the sensitivity file is "–sens". The Timer will load all the sensitivities to the map for the later usage.

For example:
Option can be something like the following:
-cell z10_test_bench –lib designs –view physical –liberty cbl250.lib –sdc z10_test_01.sdc –report critical –sens sens.txt

Experimental Results

Experiments are run to test the accuracy of the method and the correctness of the implementation and Monte Carlo size is 1000 samples.
Four benchmarks are used:

1. **z10**
2. **s382**
3. **s444**
4. **s13207**

The numbers in those four benchmarks indicate how many logic components it consists of. As one can see, the testing experiments start from fairly small scale benchmarks to large scale benchmarks. As a result, one can compare the correctness of the implementation as well as the runtime for increased scale benchmarks.

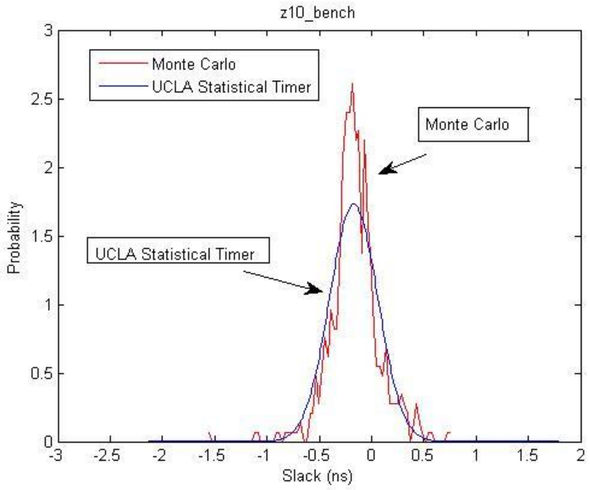
Each benchmark is run with three different types of variations:

a) **Mixed**, combined $\sigma_{global} =$ 0.07, .03, and σ_{indep} = 0.02
b) **Independent only**, with σ_{indep} = 0.02
c) **Global Variation only**, with σ_{global} = .07, .03

1. *z10_bench*

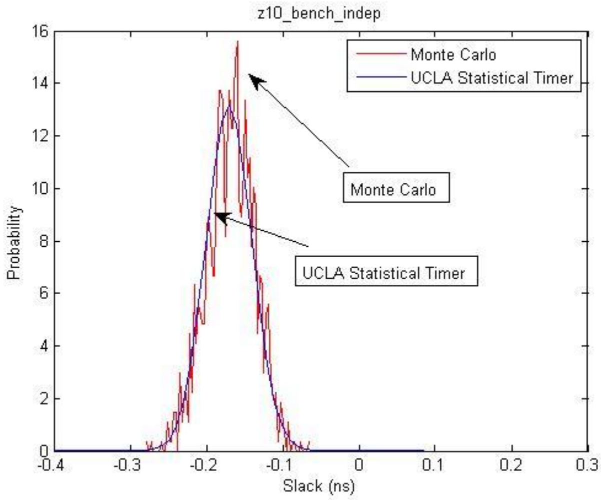
a) Mixed

Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-0.1705	-0.1595		Real Time	1.29 s	1.19 s	8.4%
Standard Deviation	0.2305	0.242		User Time	0.073 s	0.061 s	
Skew	/	-0.2268		System Time	0.045 s	0.048 s	
Kurtosis	/	3.9186					



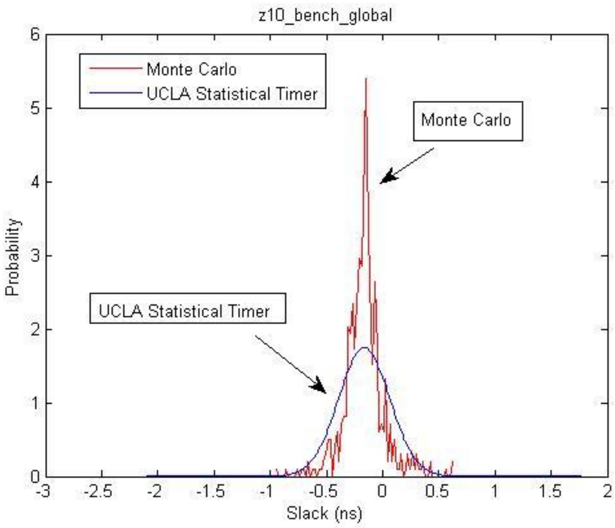
b) Independent only

Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-0.1704	-0.1679		Real Time	1.30 s	1.21 s	7.43%
Standard Deviation	0.0303	0.0312		User Time	0.062 s	0.060 s	
Skew	/	-0.1514		System Time	0.052 s	0.050 s	
Kurtosis	/	0.0365					



c) **Global only**

Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-0.1638	-0.1555		Real Time	1.22 s	1.21 s	8.26%
Standard Deviation	0.2285	0.2284		User Time	0.068 s	0.062 s	
Skew	/	-0.262		System Time	0.051 s	0.048 s	
Kurtosis	/	3.238					

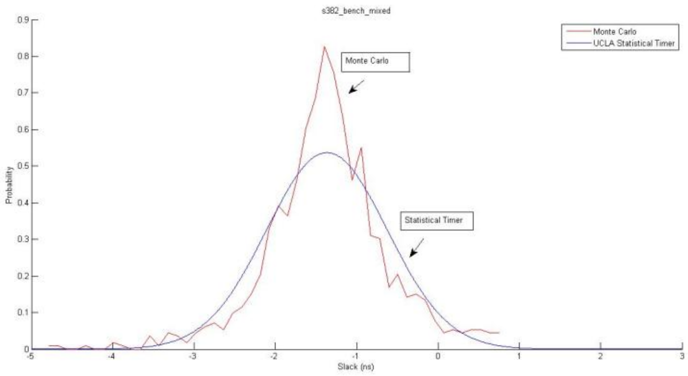


2. ***s382_bench***

a) Mixed

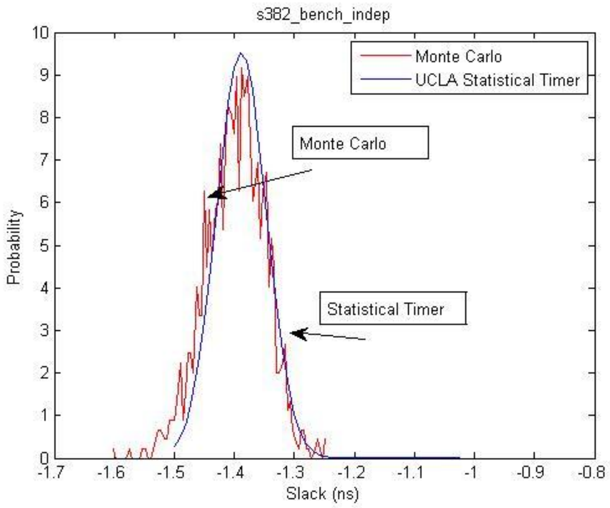
Summary Table							
	SSTA Result	Monte Carlo (500 samples)	Monte Carlo (1000 samples)		SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-1.415	-1.348	-1.366	Real Time	1.50 s	1.35 s	11.11%
Standard Deviation	0.7186	0.7564	0.744	User Time	0.155 s	0.128 s	

Skew	/	-0.700	-0.23	System Time	0.174 s	0.145 s	
Kurtosis	/	3.989	1.803				



b) **Independent only**

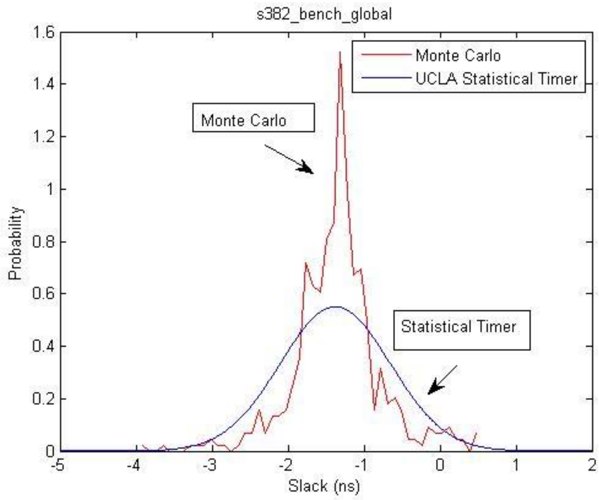
Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-1.388	-1.3984		Real Time	1.50 s	1.42 s	5.63%
Standard Deviation	0.0419	0.0432		User Time	0.182 s	0.167 s	
Skew	/	-0.23075		System Time	0.118 s	0.112 s	
Kurtosis	/	0.3179					



c) **Global only**

Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-1.378	-1.353		Real Time	1.47 s	1.41 s	4.25%
Standard Deviation	0.724	0.707		User Time	0.177 s	0.162 s	
Skew	/	-0.104		System	0.104 s	0.101 s	

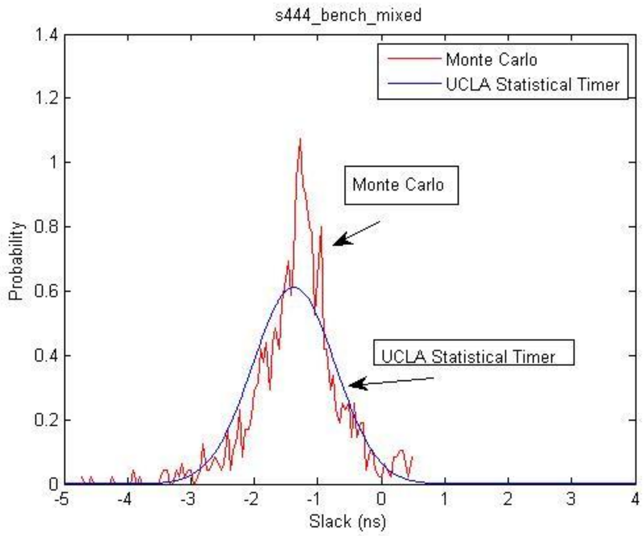
				Time			
Kurtosis	/	1.13					



3. ***s444_bench***

a) Mixed

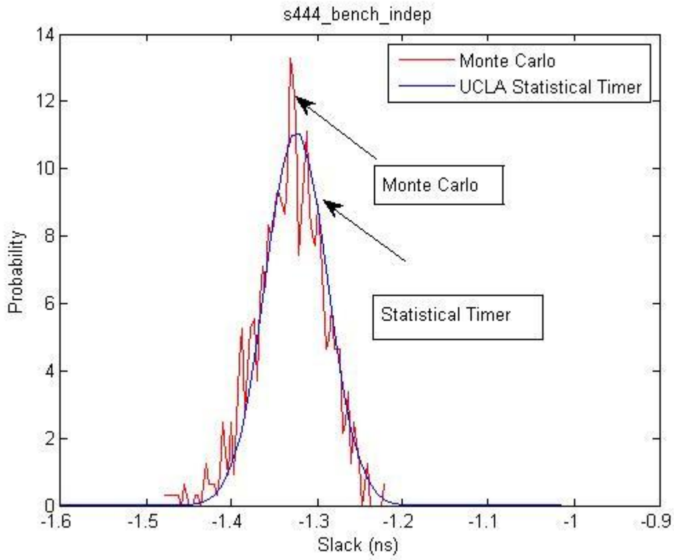
Summary Table							
	SSTA Result	Monte Carlo (500 samples)			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-1.376	-1.321		Real Time	1.94 s	1.48 s	31.08%
Standard Deviation	0.651	0.677		User Time	0.174 s	0.166 s	
Skew	/	-0.552		System Time	0.177 s	0.147 s	
Kurtosis	/	2.356					



b) Independent only

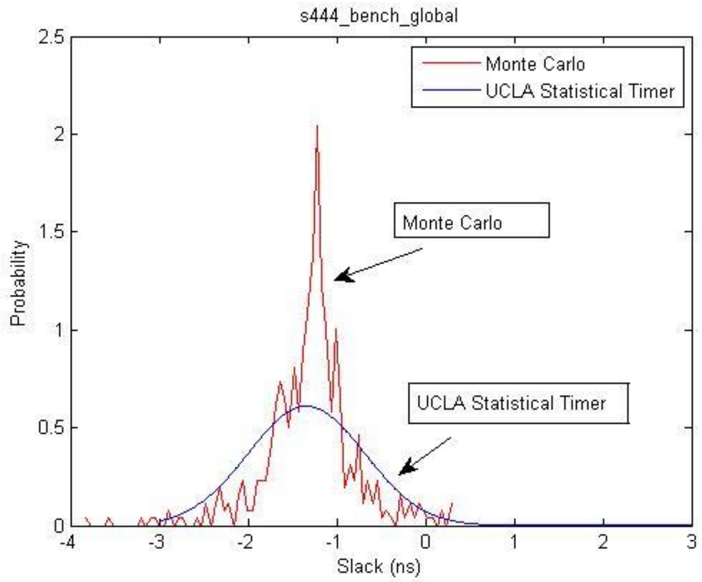
Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)

Mean	-1.324	-1.3317		Real Time	1.50 s	1.46 s	2.73%
Standard Deviation	0.0358	0.0398		User Time	0.147 s	0.142 s	
Skew	/	-0.3653		System Time	0.164 s	0.154 s	
Kurtosis	/	0.4451					



c) Global only

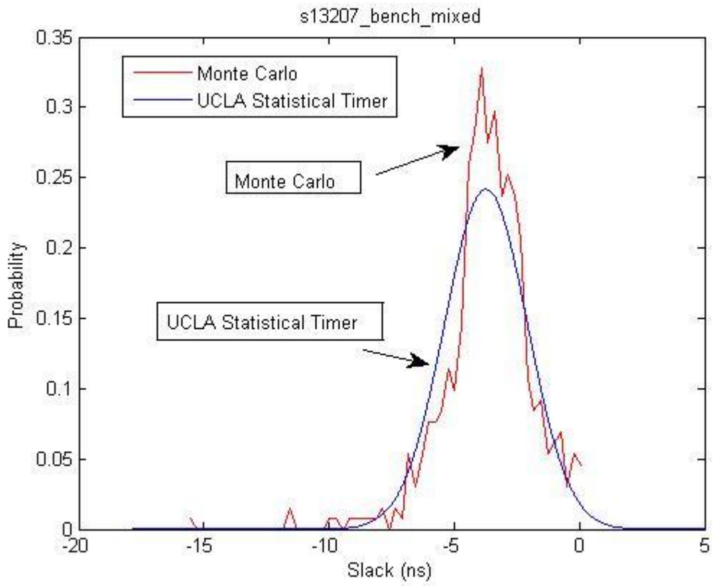
Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-1.3398	-1.318		Real Time	1.55 s	1.50 s	3.33%
Standard Deviation	0.6531	0.614		User Time	0.156 s	0.151 s	
Skew	/	-0.0904		System Time	0.88 s	0.79 s	
Kurtosis	/	2.379					



4. s13207_bench

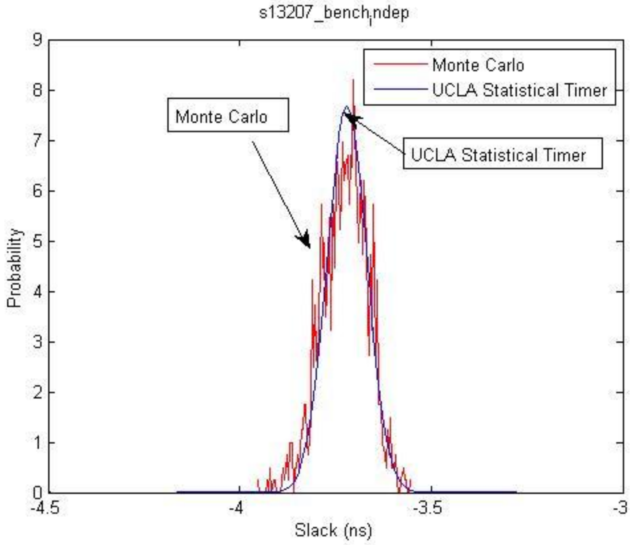
a) Mixed

Summary Table							
	SSTA Result	Monte Carlo (500 samples)			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-3.9	-3.669		Real Time	35.00 s	28.84 s	21.36%
Standard Deviation	1.637	1.771		User Time	1.661 s	1.412 s	
Skew	/	-1.23		System Time	1.319 s	1.23 s	
Kurtosis	/	5.692					



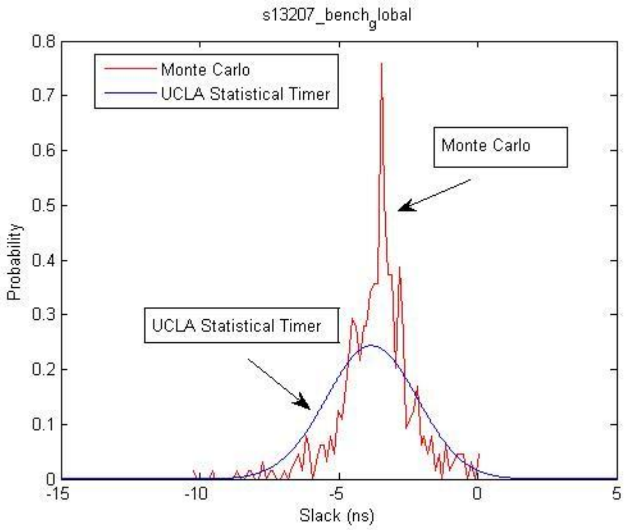
b) **Independent only**

Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-3.719	-3.726		Real Time	34.91s	32.21 s	8.38%
Standard Deviation	0.0524	0.0606		User Time	1.586s	1.437 s	
Skew	/	-0.2838		System Time	1.557 s	1.499 s	
Kurtosis	/	0.1568					



c) **Global only**

Summary Table							
	SSTA Result	Monte Carlo			SSTA RunTime	Deterministic Runtime	Difference (%)
Mean	-3.812	-3.641		Real Time	24.07 s	23.69 s	1.6%
Standard Deviation	1.637	1.419		User Time	1.58 s	1.45 s	
Skew	/	-0.668		System Time	1.38 s	1.29 s	
Kurtosis	/	3.17					



<u>Summary</u>

The experimental results and data indicate that the Gaussian distribution gotten from UCLA Statistical Timer is pretty close to the Monte Carlo results, especially for the graphs of "independent sensitivities only". Even though the mean and standard deviation gotten from the UCLA Statistical Timer and Monte Carlo are very close, the graphs of mixed sensitivities (consists of both global and independent sensitivities) and the graphs of "global sensitivities only" always have a peak around their mean. It may be caused by our small Monte Carlo sample size. For the future work, we will definitely try larger Monte Carlo sample size to see how the distribution may fit better.

References

[1] Visweswariah, C.; Ravindran, K.; Kalafala, K.; Walker, S.G.; Narayan, S.; Beece, D.K.; Piaget, J.; Venkateswaran, N.; Hemmett, J.G.; , "First-Order Incremental Block-Based Statistical Timing Analysis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.25, no.10, pp.2170-2180, Oct. 2006