Paper ID: 195

SD-SSTA: Statistical Static Time Analysis Algorithm Considering Skewed Distribution

Fuxing Deng[1*],Yihang Feng[1*], Dan Niu[2], Xiao Wu[3] and Zhou Jin[1]

1.Super Scientific Software Laboratory, China University of Petroleum-Beijing, Beijing, China

2.School of Automation, Southeast University, Nanjing, China

3.Huada Empyrean Software Co. Ltd, Beijing, China

Email: fuxing.deng@student.cup.edu.cn

2024.05.13

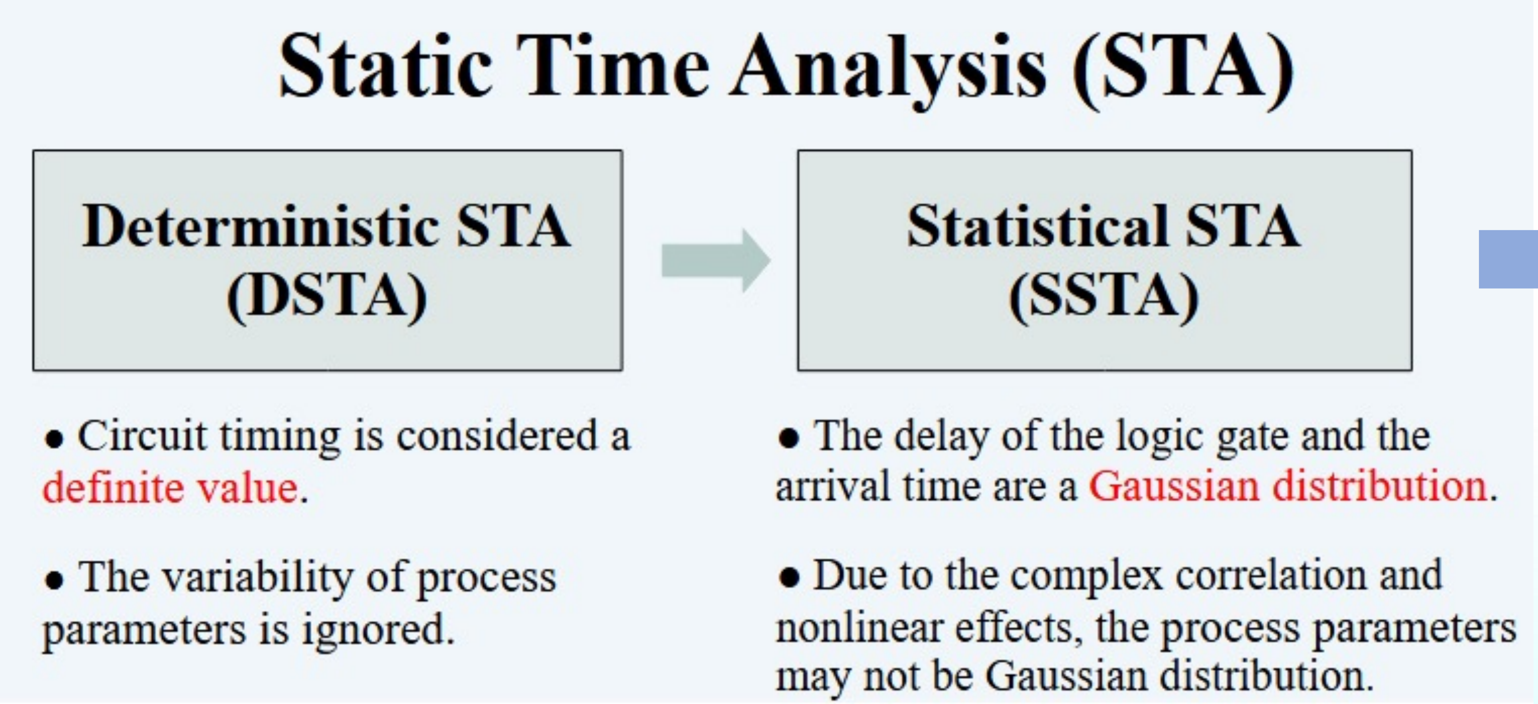
Contents

01

Background

Static Time Analysis

If the timing violation of build time or hold time exists in the integrated circuit, the actual working speed of the integrated circuit will not meet the design requirements, and even produce functional errors leading to design failure. That's why *STA is so important*.



Our method:
SSTA algorithm considering skewed distribution (SD-SSTA)

- An arbitrary probability replaces the Gaussian distribution.

Figure 1. *Development of STA.*

Static Time Analysis

The generation of skew

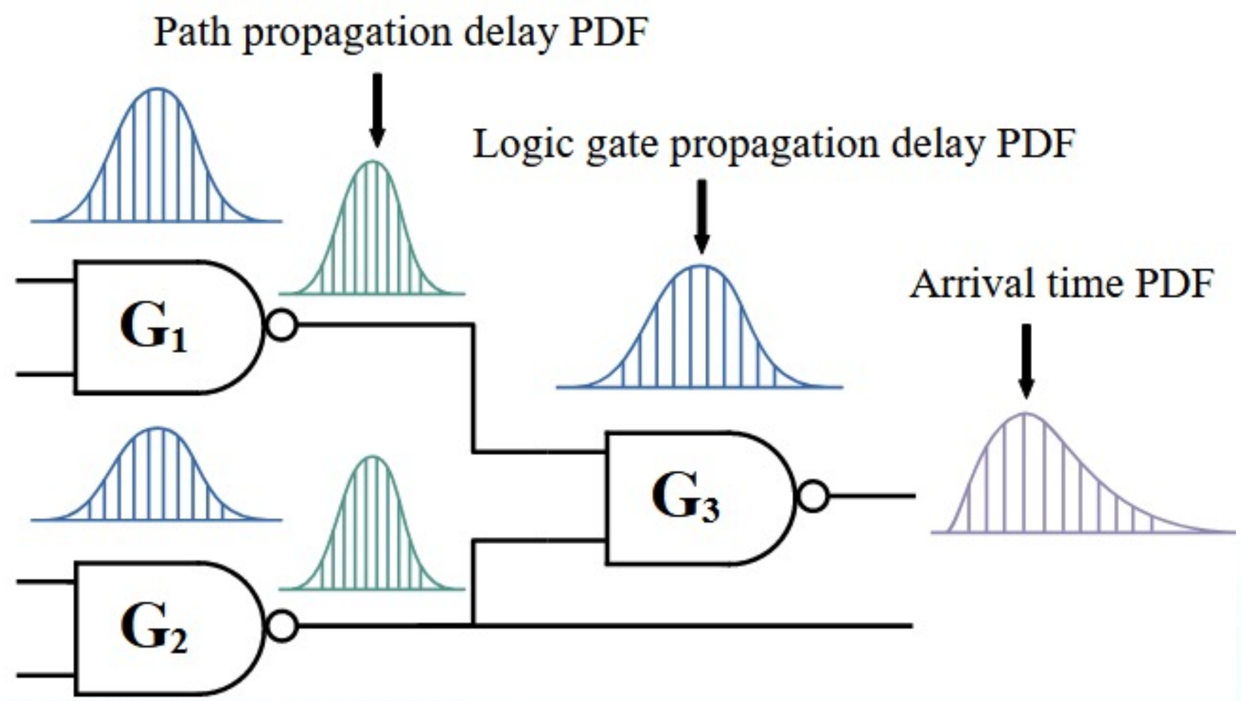


***Figure 2**. The generation of skew.*

- **Nonlinear Components**: Common nonlinear components in timing circuits, such as diodes and transistors, can introduce nonlinear responses.
- **Environmental Effects**: Changes in environmental factors such as temperature and humidity can lead to changes in component parameters.
- **Clock Skew and Jitter**: Skew and jitter in the clock signal can affect the statistical distribution of the data.

Static Time Analysis

Shortcomings of the traditional SSTA

References

[1] S. Nassif. Within-chip variability analysis. In IDEM ’ 98, 1998.

[2] S. R. Nassif. Modeling and analysis of manufacturing variations. In CICC ’01, 2001

[3] M. Imai, T. Sato, N. Nakayama, and K. Masu. Non-parametric statistical static timing analysis: An ssta framework for arbitrary distribution. In DAC ’08, 2008

- Using Gaussian distribution modeling, the results are not accurate enough
- The impact of skew is not taken into account in calculating the timing margin.
- The accuracy is lower for circuits with large variance.
- The algorithm consumes a lot of time and memory.

Our method: SD-SSTA

Challenge 1:
For large-scale circuits, the search and comparison of strings will consume a lot of memory and time.

Solve Challenge 1:
We propose **name mapping method**.

Challenge 2:
It is complicated to realize the forward propagation of non-Gaussian distribution.

Solve Challenge 2:
- We use **GMM** to model non-Gaussian.
- We **parameterize** the non-Gaussian.

Challenge 3:
Calculating the timing margin requires the effect of skewness.

Solve Challenge 3:
We introduce ***skewness coefficient*** on the basis of Gaussian distribution.

Challenge 4:
A circuit with a large standard deviation has lower accuracy.

Solve Challenge 4:
We set a **scale parameter** in the formula.

02

SSTA algorithm considering skewed distribution

Overall Framework

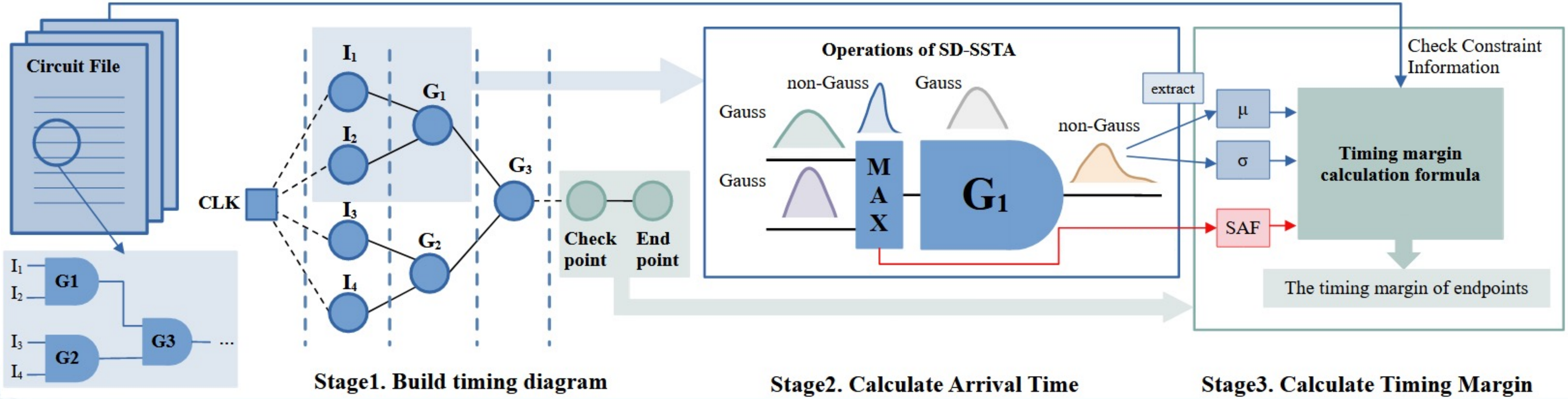


Figure 3. *Implementation process of SD-SSTA algorithm.*

SD-SSTA *Build Timing Diagram*

Read the circuit design file and establish the data structure of the timing diagram.

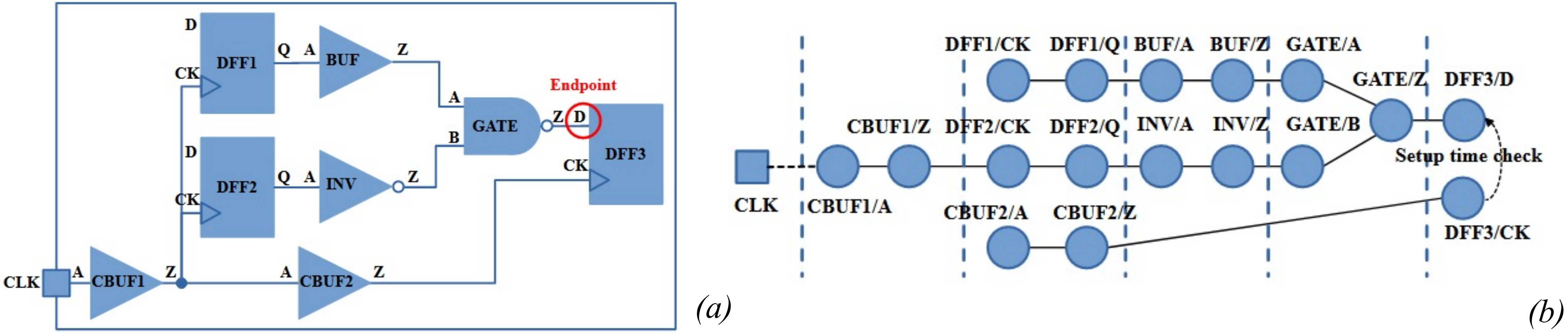


Figure 4. *Example circuit in (a) and its timing graph in (b).*

We collect statistical data of circuit elements and organize circuit design files, which include various important parts describing circuit structure and timing information. A simple example is shown in ***Figure 4***.

SD-SSTA *Build Timing Diagram*

Solve Challenge 1

We propose a ***name mapping method*** to map all names into integer numbers, and only use integer numbers for calculation during the operation of the program.

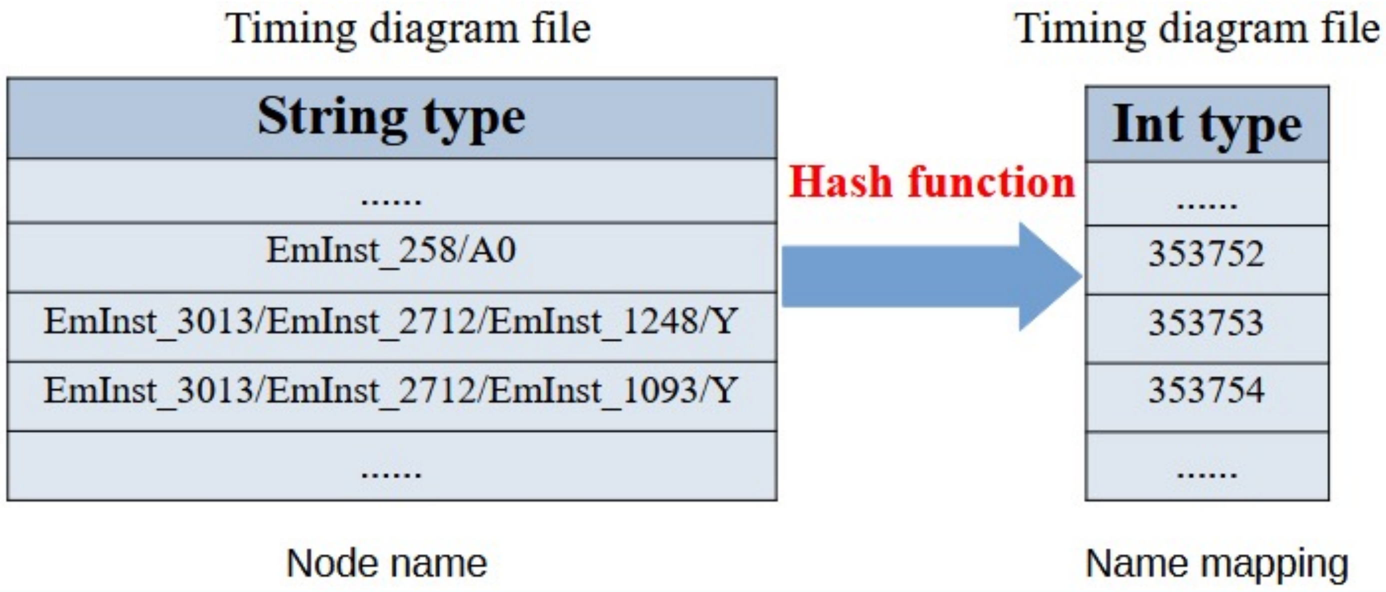


Figure 5. *The mapping process for the Hash function.*

SD-SSTA *Operations for SD-SSTA*

Solve Challenge 2

- 2-1： **Modeling non-Gaussian distribution with GMM**

A **GMM** can be regarded as a model composed of N individual Gaussian models.

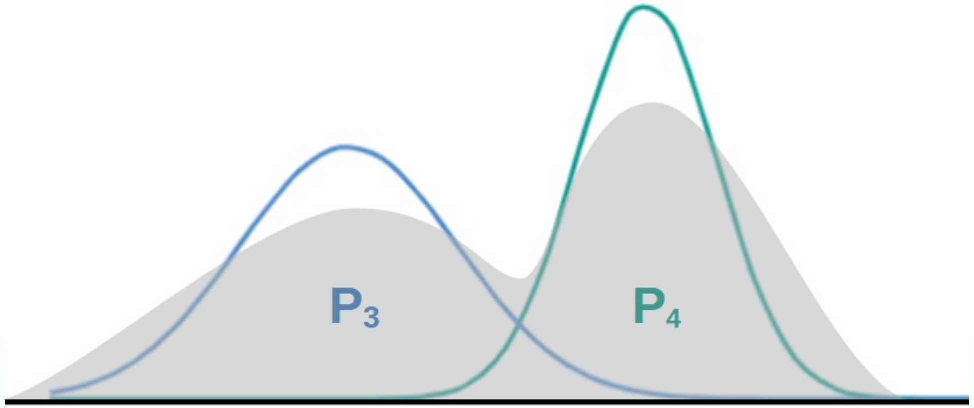


Figure 4. *A GMM consists of two Gaussian distributions.*

The mathematical form of **GMM** is as ***equation (1)***. Such basis function ϕ(x) is called ***Radial Basis Function (RBF)***.

$$f_{GMM}(x) = \sum_{i=1}^{N} \omega_i \cdot \phi(x \mid \mu_i, \sigma_i) \qquad (1)$$

Advantages of Gaussian mixture model:

- Gaussian distributions possess excellent **mathematical properties** and **computational performance**.
- **GMM** can smoothly fit PDFs of arbitrary shapes.

SD-SSTA *Operations for SD-SSTA*

Solve Challenge 2

- 2–2： **Parameterize the non-Gaussian distribution**

A non-Gaussian distribution is represented by μ, σ and ***skewness coefficient***.

• **calculate μ and σ:** We propose to model the PDF of the max with **GMM**. It can be decomposed into ***RBFs***. Furthermore, we extract the μ and σ of each fitted ***RBF*** from the model's attributes.

$$\mu_{\text{gate}} = \sum_{i=1}^{N} \omega_i \cdot \mu_i \quad \text{and} \quad \sigma_{\text{gate}} = \sqrt{\sum_{i=1}^{N} \omega_i \cdot \sigma_i^2} \tag{2}$$

SD-SSTA *Operations for SD-SSTA*

Solve Challenge 2

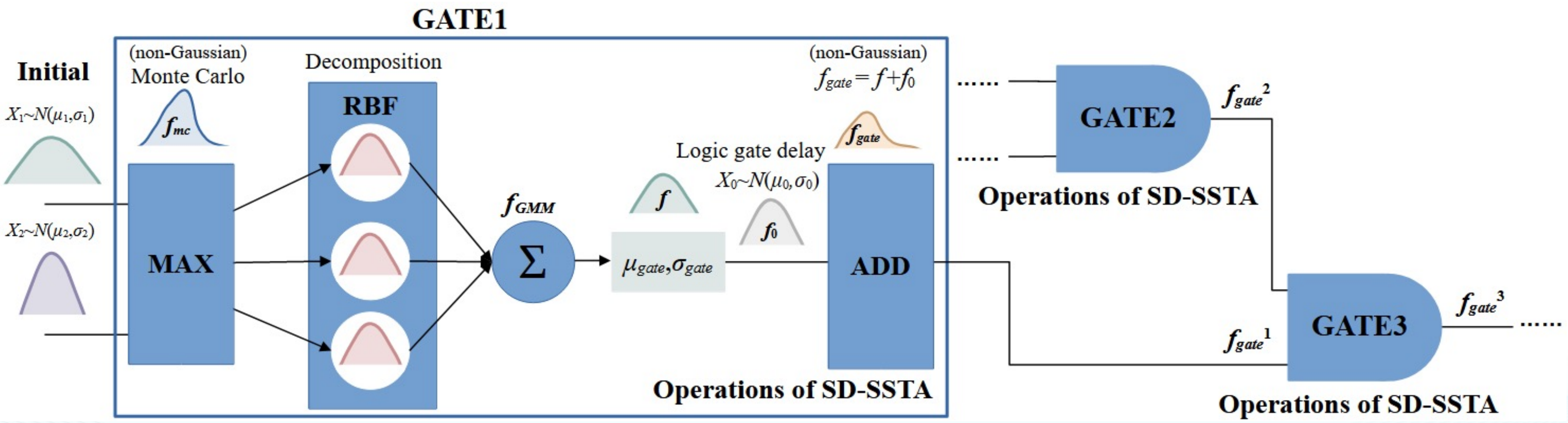


Figure 6. *Implementation of non-Gaussian distributed Max and Add operations.*

SD-SSTA *Operations for SD-SSTA*

Solve Challenge 3

We introduce ***skewness coefficient*** on the basis of Gaussian distribution.

- **calculate *skewness coefficient*** (***skew*** in ***equation (4)***)**:** By calculating the third moment, we can describe the skewness of a non-Gaussian distribution after the ***Max operation.***

$$f_{\max}(x,\rho)=f_1(x)\Phi\left[\frac{1}{\sqrt{1-\rho^2}}\left(\frac{x-u_2}{\sigma_2}-\rho\frac{x-u_1}{\sigma_1}\right)\right]+f_2(x)\Phi\left[\frac{1}{\sqrt{1-\rho^2}}\left(\frac{x-u_1}{\sigma_1}-\rho\frac{x-u_2}{\sigma_2}\right)\right] \tag{3}$$

$$skew=E\left[(X-\mu)^3\right]=\int_{-\infty}^{\infty}(x-\mu)^3 f_{\max}(x)dx \tag{4}$$

SD-SSTA *Calculation of Time Margin*

Solve Challenge 4

We introduce ***SAF*** into the formula for calculating the timing margin. The value of ***SAF*** is affected by two factors, one is the skewness and the other is the variance of the delay.

$$\boldsymbol{SAF} = skew \times k \tag{5}$$

$$Timing\ margin = reqTime - arrTime + \boldsymbol{SAF} \tag{6}$$

For different circuits, we dynamically adjust the scale parameter k in ***SAF*** according to the variance size of the relevant circuit design file.

03

Performance

Environment Setup

● Environment Configuration

CPU	AMD EPYC 7702 64-Core Processor
System	Ubuntu
Architecture	x86_64
IDE	python

Table 1. *Environment configuration.*

● Test cases

We use three test cases for the experiment, each case containing three circuit design files:
(1) Timing diagram file
(2) Setuptime check file
(3) Endpoints list
The information of different cases is shown in ***Table 2***.

	Circuit 1	Curcuit 2	Curcuit 3
Timing diagram file	7894	8794	126998
Endpoints list	1313	1313	4502
Mean of σ	0.004047	0.036749	0.000744

The number of nodes and edges in the timing diagram.

Table 2. *Information about the test cases.*

Environment Setup

- **Evaluation Index**

We use the results of ***equation (7)***, $score_{total}$, as an evaluation index. It can measure the accuracy of SD-SSTA algorithm to calculate the timing margin of all endpoints.

$$score_{total} = 100 \times \frac{\sum_{all\,results} score_{single}}{number\ of\ results} \tag{7}$$

$$score_{single} = \begin{cases} 1 - err_{single} & , err_{single} \leq 20\% \\ 0 & , err_{single} \leq 20\% \end{cases} \tag{8}$$

$$err_{single} = abs\left(\frac{our\ results - standard\ result}{standard\ result}\right) \tag{9}$$

equation (7): The total score of all endpoints. (full score are 100.)

equation (8): The score of a single endpoint. (When the error of a single endpoint exceeds 20%, the result is invalid, and the endpoint score is zero.)

equation (9): The error rate of a single endpoint.

Results

- ***Accuracy result***

Table 3 compares the timing margin calculated by conventional **SSTA** and our proposed **SD-SSTA** algorithms for three different test cases.

Method	Accuracy score of different cases		
	Circuit 1	Circuit 2	Circuit 3
SSTA	97.450931	80.584524	99.425688
SD-SSTA	99.477106	98.251913	99.861775

Table 3. *Final score for each case.*

- SD-SSTA algorithm has **higher accuracy** than the traditional SSTA.
- The variance of circuit 1 and circuit 3 is too small, so the traditional SSTA also can achieve an accurate calculation.
- The variance of circuit 2 is too large, the traditional SSTA lead to **high error** and the **superiority** of SD-SSTA algorithm becomes apparent.

Results

- *Performance Analysis*

We compare the algorithm using ***name mapping method*** with the original algorithm, and the efficiency of optimization is shown in ***Figure** 7*.

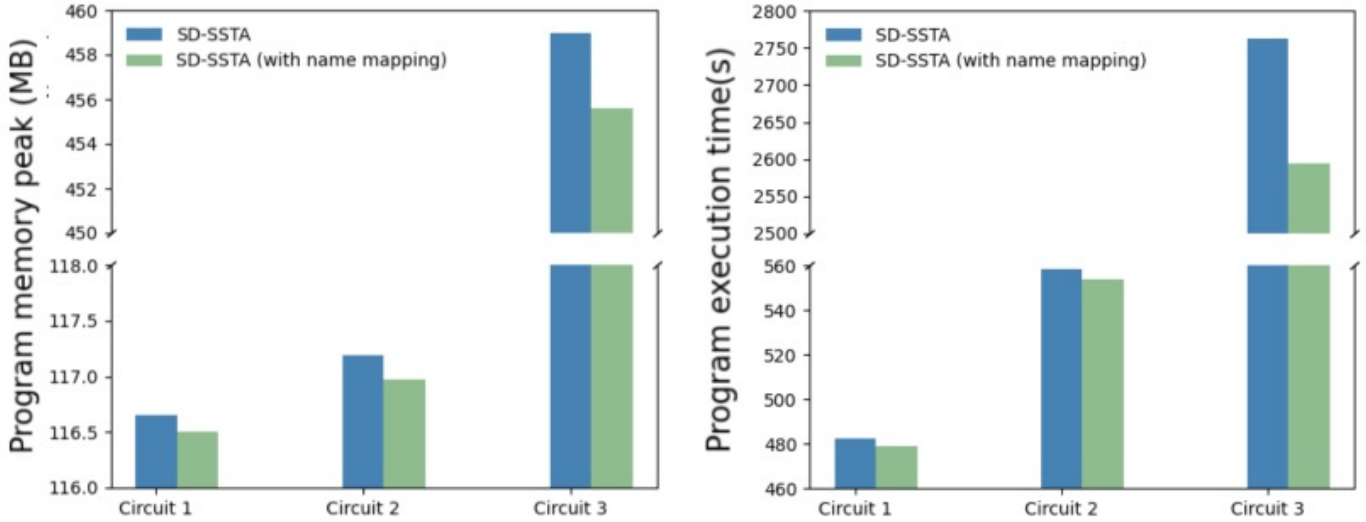


***Figure** 7. Time and memory comparison before and after the name mapping method is used.*

- Our proposed ***name mapping method*** reduces the time and memory of the algorithm.
- Its effect is especially obvious for **large-scale circuits** (Circuit 3).

04

Conclusion

Conclusions

A non-Gaussian distribution based SSTA method, **SD-SSTA**, is proposed.

- We use **GMM** to model non-Gaussian distribution.
- A non-Gaussian distribution is represented by $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and ***skewness coefficient***, that is, the non-Gaussian distribution is parameterized.
- In the calculation of timing margin, we consider the influence of **skew** and introduce ***SAF*** into the formula.
- The memory and time performance of the algorithm is improved by **name mapping**.

Thanks for listening!