

# A Performance Optimization Method by Gate Sizing using Statistical Static Timing Analysis

Masanori Hashimoto

Dept. Communications & Computer Engineering  
Kyoto University  
Sakyo-ku, Kyoto, 606-8501, Japan  
hasimoto@vlsi.kuee.kyoto-u.ac.jp

Hidetoshi Onodera

Dept. Communications & Computer Engineering  
Kyoto University  
Sakyo-ku, Kyoto, 606-8501, Japan  
onodera@vlsi.kuee.kyoto-u.ac.jp

## ABSTRACT

We propose a gate resizing method for delay and power optimization that is based on statistical static timing analysis. Our method focuses on the component of timing uncertainties due to local random fluctuation. Utilizing our method, over-design of a circuit can be eliminated and high-performance and high-reliability LSI design can be realized. The effectiveness of our method is examined by 6 benchmark circuits. We verify that our method can reduce delay and power dissipation from the circuits optimized without the consideration of fluctuation.

## 1. INTRODUCTION

There are several sources that cause the uncertainties of circuit delay time, such as manufacturing fluctuation, supply voltage and temperature change, estimation error of wire capacitance and resistance, uncertainties of wire capacitance during physical design, diversity in signal waveforms, and so on. These sources can be classified into two categories. The first category is a global change that applies to all gates and wires similarly in a certain region. The second category is a random change that indicates a certain statistical distribution. As for the global change, there is a traditional and widely-used method to consider the delay time uncertainties. In this method, three values (best/typical/worst-case values) are prepared for the delay time of each gate and wire. Then the circuit delay time is calculated using each-case value for purpose by purpose. This is a reasonable approach for the global change.

On the other hand, the random change is not well considered in LSI design. Due to the random change, the delay time of each gate and wire has a certain probability distribution. In one case, a certain amount of design margin is set to avoid the effect of the delay time uncertainties by the random change. In this method, the decision of the design margin is difficult, which results in excessive design margin and over-design of the circuits. In another case, the delay time of each gate and wire is defined as the worst-case value, for example,  $\text{mean}+3\sigma$ . In this case, the estimated delay time of a critical path is pessimistic, and the delay of the shortest path can not be considered. Therefore, in order to design a circuit with high confi-

dence and eliminate over-design, a statistical static timing analysis method and a circuit optimization method considering the random change are necessary.

We propose a performance optimization method considering the random change based on statistical timing analysis. As for statistical timing analysis, there are several proposals [1–5]. The methods proposed in Refs. [1–3] are Monte Carlo simulation-based techniques, so these methods are not suitable for large circuits from the point of computation time. The method proposed by Berkerlaar in Ref. [4, 5] is based on a static timing analysis method. This method does not require any simulations, and the complexity of the timing analysis is linear to the circuit scale. So the timing analysis can be done in a realistic computation time. Although this method works well for the estimation of the mean delay, it underestimates the worst delay (eg.,  $\text{mean}+3\sigma$ ) [4]. In a statistical analysis, it is important to estimate a worst case value. We therefore devise a technique for accuracy improvement and utilize this method for performance optimization.

In the case of the performance optimization based on statistical static timing analysis, slack [6], which represents the timing criticality at each gate and is widely used for performance optimization under deterministic delay model, can no longer be a useful measure under statistical environment. We therefore propose a new measure “criticality” that represents the timing criticality at each gate, and device performance optimization algorithms utilizing the “criticality”. In Ref. [5], the gate sizing problem is formulated as a nonlinear programming problem, where the objective function and the constraints are expressed as analytic forms. In this method, the delay should be represented in a simple analytical equation, which degrades the accuracy of the delay calculation. On the other hand, our method can utilize any gate/wire delay calculation methods.

Our performance optimization method has various applications, such as uncertainties of wire capacitance during physical design, local fluctuation in transistor characteristics, local variation of supply voltage and temperature, and so on. The proposed performance optimization method can eliminate over-design of a circuit and contribute high-performance and high-reliability LSI design.

This paper is organized as follows. Section 2 discusses the statistical static timing analysis method. Section 3 explains the performance optimization algorithms of gate sizing. Section 4 discusses some applications of our performance optimization method. Section 5 demonstrates some experimental results. Finally, Section 6 concludes the discussion.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD 2000, San Diego, CA.

Copyright 2000 ACM 1-58 113-191-7/00/0004.. \$5.00

## 2. STATISTICAL STATIC TIMING ANALYSIS

In this section, a statistical timing analysis method is discussed. We first explain the basic concept of the statistical static timing analysis proposed in Ref. [4]. Next, we discuss an approximation method of the delay distribution used in the statistical static timing analysis. We then propose a new measure ‘‘criticality’’ that represents the timing criticality at each gate.

### 2.1 Static Timing Analysis

We first explain a conventional(not statistical) static timing analysis method briefly. Suppose a gate that has  $n$ -input and 1-output ports(Fig. 1).  $T_i$  is the latest arrival time of signals at the  $i$ -th input.  $t_i$  is the gate delay time from the  $i$ -th input to the output.  $T_i$  and  $t_i$  have different values for rise and fall transitions. In Section 2, we do not distinguish rise/fall transitions for simplifying the explanation. But the real implementation in Section 5 considers the delay difference for rise/fall transitions. The latest arrival time of the signal transitions at the output,  $T_{out}$ , is represented as follows.

$$T_{out} = \max_{i=1}^n (T_i + t_i). \quad (1)$$

Using Eq. 1, the latest arrival time at each gate can be calculated incrementally without tracing all paths.

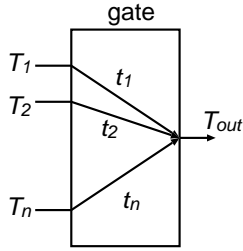


Figure 1: Gate Delay Model

### 2.2 Statistical Static Timing Analysis

In a conventional static timing analysis, each delay time of gates and wires is a constant value. On the other hand, under the existence of uncertainties in circuit delay time, each delay time is not a constant and it has a statistical distribution, which is considered for delay calculation in the statistical static timing analysis. The basic concept of the statistical static timing analysis has been proposed in Ref. [4]. We first explain this method briefly. Next, we discuss a technique that improves the accuracy of the timing analysis.

We model the distribution of the latest signal arrival time at the  $i$ -th input as a normal distribution of a stochastic variable  $T$  with mean  $\mu_{T_i}$  and standard deviation  $\sigma_{T_i}$ . We also assume that the gate delay time from the  $i$ -th input to the output is distributed with a stochastic variable  $t$ , mean  $\mu_{t_i}$  and standard deviation  $\sigma_{t_i}$ .

Here, Eq. 1 is converted for the statistical timing analysis. We define the probability density function  $f_i$  that corresponds to the distribution of  $T_i+t_i$ . The mean and standard deviation of  $T_i+t_i$  are represented as  $\mu_{T_i} + \mu_{t_i}$  and  $\sqrt{\sigma_{T_i}^2 + \sigma_{t_i}^2}$  respectively. The probability density function  $F_i$  is defined as follows.

$$F_i(x) = \int_{-\infty}^x f_i(\chi) d\chi. \quad (2)$$

As an example of statistical max operation, we take up  $C = \max(A, B)$ , with stochastic variables  $A$ ,  $B$  and  $C$ . In this case,

the following relation holds at any  $x$ .

$$P(C \leq x) = P((A \leq x) \cap (B \leq x)), \quad (3)$$

where  $P(Condition)$  represents the probability that  $Condition$  is satisfied. When the statistical correlation between  $A$  and  $B$  is ignored, Eq. 3 can be transformed as follows.

$$P(C \leq x) = P(A \leq x) \cdot P(B \leq x). \quad (4)$$

We define the probability density functions of  $A$ ,  $B$  and  $C$  as  $f_A$ ,  $f_B$  and  $f_C$ . Eq. 4 can be expressed as follows.

$$\int_{-\infty}^x f_C d\chi = \int_{-\infty}^x f_A d\chi \cdot \int_{-\infty}^x f_B d\chi. \quad (5)$$

Differentiating Eq. 5, the following equation can be obtained.

$$f_C = f_A(x) \cdot \int_{-\infty}^x f_B d\chi + f_B(x) \cdot \int_{-\infty}^x f_A d\chi. \quad (6)$$

Eq. 6 can be rewritten as follows.

$$P(C = x) = P(A = x) \cdot P(B \leq x) + P(B = x) \cdot P(A \leq x). \quad (7)$$

Extending Eq. 6 for  $n$  stochastic variables, the probability function  $f_{out}$ , which corresponds to the distribution of the latest arrival time  $T_{out}$ , can be represented as follows.

$$f_{out}(x) = \sum_i^n \left[ f_i(x) \cdot \prod_{j \neq i} F_j(x) \right]. \quad (8)$$

The probability density function of the overall circuit delay time can be obtained by applying the probability density function at each primary output to  $f_i$ .

The distribution of the latest arrival time,  $f_{out}$ , is different from a normal distribution, though assumed to be normal. In Ref. [4], the distribution shape of  $f_{out}$  is discussed under some conditions with the result that the distribution of  $f_{out}$  is approximated to a normal distribution. From now, we discuss the extraction method of mean and standard deviation. Figs. 2 and 3 show an example of the approximation to the normal distribution.  $f_{out}$  represents Eq. 8 under the following conditions. The mean and standard deviation of  $f_1$ , the mean and standard deviation of  $f_2$  and  $n$  are 0, 1, 0.6, 0.6 and 2 respectively. The curve falls slower than it rises. We take up two approximation method of  $f_{out}$  to a normal distribution.

**Method 1** Using Monte Carlo technique, extract the values of the mean  $m$  and the standard deviation  $\sigma$ .

**Method 2** Find the values of  $x_0$  and  $x_1$  that satisfy Eqs. 9 and 10. The mean  $m$  is calculated as  $(x_0 + x_1)/2$  and the standard deviation  $\sigma$  is  $(x_1 - x_0)/6$ .

$$0.013499 = \int_{-\infty}^{x_0} f_{out}(x) dx. \quad (9)$$

$$0.013499 = \int_{x_1}^{\infty} f_{out}(x) dx. \quad (10)$$

Method 1 is adopted in Ref. [4]. Method 2 adjusts the value of the integral of  $f_{out}$  from  $m + 3\sigma$  to infinity and the integral from negative infinity to  $m - 3\sigma$  using numerical integration. At a glance, Method 1 approximates  $f_{out}$  to a normal distribution better than Method2(Fig. 2). In the case that we estimate the circuit delay around the point of  $m + \sigma$ , Method1 is surely suitable for the approximation. But, when the circuit delay is defined as  $x_1$  in

Eq. 10, where  $x_1$  corresponds to  $m + 3\sigma$  of the normal distribution, Method1 underestimates the delay time(Fig. 3). Though the delay time  $x_1$  derived from  $f_{out}$  is 3.00,  $x_1$  of Method 1 is 2.64. This underestimation is caused by the distribution shape difference between  $f_{out}$  and Method 1 in the region of  $x > 3$ (Fig. 3). From the definition, Method 2 can calculate  $x_1$  accurately.

In order to calculate  $x_1$  accurately, the approximation of  $f_i$  where  $x$  is larger than  $x_1$  is important. The value  $x_1$  is nearby or larger than  $m + 3\sigma$  of each  $f_i$ . In Method 2, the distribution shape of  $f_{out}$  where  $x$  is larger than  $m + 3\sigma$  is well approximated(Fig. 3), which contributes the accurate calculation of  $x_1$  at the fan-out gates that the gate drives.

From above discussion, in the case that the circuit delay is defined as  $x_1$  of Eq. 10, i.e.  $m + 3\sigma$  of the normal distribution, Method2 is good for the approximation. Hereafter, we define the circuit delay as the above definition. When the delay time is evaluated at the other point, such as  $m + 2\sigma$  and  $m + 4\sigma$ , we change the value of the left term in Eqs. 9, 10 into 0.022750 and 0.000031671 respectively.

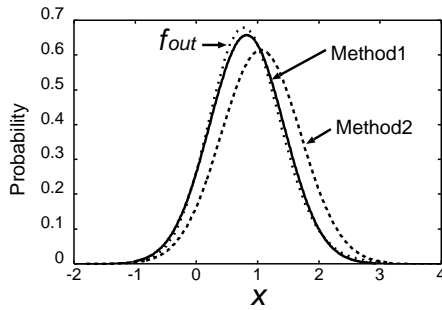


Figure 2: Approximation to Normal Distribution

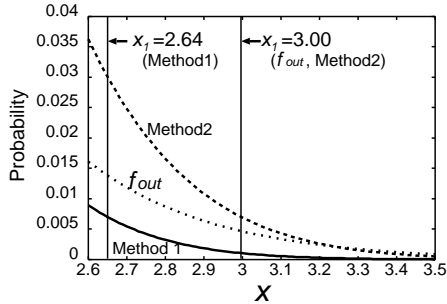


Figure 3: Approximation to Normal Distribution(Magnified)

### 2.3 Criticality

In the case of a conventional(not statistical) static timing analysis method, slack is a useful measure that represents the timing criticality at each gate[6]. Many optimization algorithms using slack have been proposed[7–9], and slack helps to reduce the computation time required for the optimization considerably. But in the statistical static timing analysis, slack can not be used as a measure of timing criticality. Since slack is defined as the time difference between the required arrival time and the latest arrival time, the required arrival time at each gate is computed from the primary outputs. In statistical static timing analysis, the required arrival time at each input can not be calculated independent of the arrival times at the other inputs. It is because the arrival time at the output is affected by all the inputs' arrival time(Eq. 8). Thus, the required arrival time can not be propagated. Also the combination of the mean  $m$  and the standard deviation  $\sigma$  at each gate, which satisfies

the delay constraint, is not determined uniquely. So, the required arrival time can not be defined. We therefore introduce a new measure “criticality” that represents the timing criticality at each gate. The term in the bracket of Eq. 8 represents the following probability.

$$f_i(x) \cdot \prod_{j \neq i} F_j(x) = P(T_i + t_i = x) \cdot \prod_{j \neq i} P(T_j + t_j \leq x). \quad (11)$$

The input with the high probability of Eq. 11 affects the distribution of  $T_{out}$  at  $x$  strongly. The probability of Eq. 11 expresses the magnitude of the influence that the  $i$ -th input gives to  $f_{out}$  at  $x$ . We define “influence $_i$ ” that represents the influence proportion of the  $i$ -th input in the range of  $x \geq x_1$  as follows.

$$influence_i = C_1 \cdot \int_{x_1}^{\infty} f_i(x) \cdot \prod_{j \neq i} F_j(x) \cdot exp(C_2 \cdot x) dx, \quad (12)$$

where  $C_1$  is a normalization coefficient to satisfy  $\sum_i influence_i = 1$  and  $C_2$  is a constant. A term  $exp(C_2 \cdot x)$  is multiplied in order to emphasize the region of large arrival time. When  $influence_i$  is 1,  $f_{out}$  in  $x \geq x_1$  is determined by the  $i$ -th input and the other inputs do not affect  $f_{out}$ . Conversely, when  $influence_i$  is 0, the  $i$ -th input does not influence on  $f_{out}$  in  $x \geq x_1$  at all. “Influence” at each primary output on the overall circuit delay time can be similarly obtained by applying the probability density function at each primary output to  $f_i$ .

We now explain how to calculate “criticality” that represents the timing criticality at each gate. “Criticality” at each gate is defined as the amount of the contribution to the circuit delay by the paths that go through the gate. We propagate “criticality” from primary outputs to primary inputs. Suppose Fig. 4 given for an example.  $i(G)$  is defined such that the  $i(G)$ -th input is connected with gate  $G$ . A term  $influence_{i(G)}(G_j)$  means how much the  $i(G)$ -th input affects the timing at gate  $G_j$  in  $x \geq x_1$ . In other words,  $influence_{i(G)}(G_j)$  represents how easily the timing criticality propagates from gate  $G_j$  to gate  $G$ . Therefore “criticality” propagated from gate  $G_j$  to gate  $G$  is represented as  $influence_{i(G)}(G_j) \cdot criticality(G_j)$ .

$$criticality(G) = \sum_j^m influence_{i(G)}(G_j) \cdot criticality(G_j), \quad (13)$$

where  $m$  is the number of fan-outs for gate  $G$ . At primary outputs, “influence” means the timing criticality itself. It is because the primary output with large “influence” affects the circuit delay strongly, i.e. the timing criticality is high. So, “criticality” at primary outputs is set to 1, which enables that Eq. 13 is hold even when  $G_j$  is a primary output. We can calculate “criticality” by the breadth-first trace from the primary outputs.

The complexity of this statistical timing analysis method and the calculation of “criticality” is linear to the circuit scale. This property of the complexity make it possible to estimate and optimize the circuit delay of a large circuit.

### 3. OPTIMIZATION ALGORITHM

In this section, we explain a performance optimization algorithm based on statistical static timing analysis by gate resizing. We show two algorithms, one is for delay optimization and the other is for power(area) optimization. These algorithms utilizes “criticality” explained in the previous section.

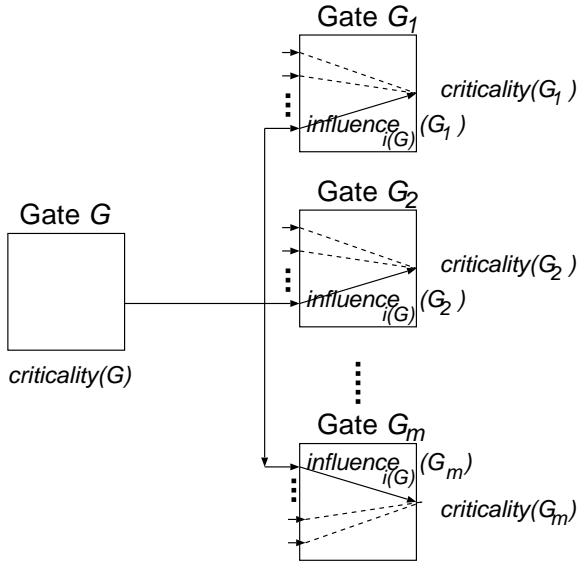


Figure 4: Propagation of ‘‘Criticality’’

### 3.1 Delay Optimization

The delay optimization algorithm is shown below.

- Step 1: put all gates into list  $L$ .
- Step 2: if  $L$  is empty or delay constraint is satisfied, finish optimization.
- Step 3: find the gate with maximum criticality in  $L$ .
- Step 4: resize the gate.
- Step 5: if delay does not decrease, cancel Step 4 and remove the gate from list  $L$  and go back to Step 2.
- Step 6: go back to Step 1.

We first put all gates into the list  $L$  of the resizing candidate. When the candidate list  $L$  is empty or the delay constraint is satisfied, the optimization process finishes. We find the gate with maximum criticality in  $L$ . It is because the gate with large criticality affects the circuit delay time strongly. The found gate is resized. If the circuit delay does not decrease, we cancel the resize operation and remove the gate from  $L$  and go back to Step 2. Otherwise, we go back to Step 1.

### 3.2 Power(Area) Optimization under Delay Constraint

We explain the gate resizing algorithm for power(area) reduction.

- Step 1: put all gates into list  $L$ .
- Step 2: if  $L$  is empty, finish optimization.
- Step 3: find the gate with minimum criticality in  $L$ .
- Step 4: resize the gate down.
- Step 5: if delay violates, cancel Step 4 and remove the gate from list  $L$  and go back to Step 2.
- Step 6: go back to Step 1.

We first put all gates into the list  $L$  of the resizing candidate. When the candidate list  $L$  is empty, the optimization process finishes. We find the gate with minimum criticality in  $L$ , because the gate with small criticality scarcely influence on the circuit delay. The found

gate is down-sized. If the delay constraint is violated, we cancel the resize operation and remove the gate from  $L$  and go back to Step 2. Otherwise, we go back to Step 1.

The optimization algorithm explained above has the possibility of falling into a local minimum solution. In order to escape from a local minimum solution, we optimize the circuit delay a little bit. After that, we apply the above algorithm again. We repeat this loop for several times.

## 4. APPLICATIONS

In this section, we show some applications of the statistical timing analysis method and the optimization algorithm explained in previous sections. Performance optimization based on the statistical timing analysis has a considerable possibility to contribute high-performance and high-reliability LSI design.

### 4.1 Uncertainties of Wire Capacitance during Physical Design and Uncertainties in Signal Waveforms

As the influence of wire on the circuit delay increases, timing closure has become a serious problem. This problem is caused by the uncertainties of wire capacitance during physical design. Also, the wire capacitance estimated from a final layout has a certain amount of errors. Because of the simple definition of the transition time, there are many different waveforms that have the same transition time, which causes the gate delay uncertainty.

When the gate delay is derived from the two dimensional look-up table with capacitive load and transition time as parameters, the gate delay is represented as follows.

$$delay = a_0 + a_1 \cdot t_{tran} + a_2 \cdot c_{load} + a_3 \cdot t_{tran} \cdot c_{load}, \quad (14)$$

where  $a_0, a_1, a_2$  and  $a_3$  are the constants decided by the look-up table,  $c_{load}$  is the load capacitance and  $t_{tran}$  is the transition time of the input signal. If the uncertainties of  $c_{load}$  at each design phase and  $t_{tran}$  can be modeled properly, the distribution of the gate delay can be derived. Then, the proposed performance optimization method can eliminate the excessive design iteration and the over-design.

### 4.2 Local Fluctuations in Transistor Characteristics, Supply Voltage and Temperature

The local variation of the transistor characteristics is represented as the fluctuation of the device parameters( $v_i, \beta, \dots$ ) and the process parameters( $t_{ox}, W, L, \dots$ ). The operating parameters( $V_{DD}, Temp$ ) also fluctuate locally. The gate delay time  $delay$  can be represented as a function of  $p_i$ , where  $p_i$  corresponds to each device, process, or operating parameters. When the local changes are not so large, the change of the gate delay time  $\delta delay$  can be represented as follows.

$$\delta delay = \sum_i d_i \cdot \delta p_i, \quad (15)$$

where  $d_i$  is a constant. In the case of the local fluctuation,  $\delta p_i$  varies according to a certain statistical distribution. The distribution of the gate delay time can be obtained. With the derived delay distribution, we can optimize the circuits considering the local fluctuations.

## 5. EXPERIMENTAL RESULTS

In this section, we show some experimental results. We first verify the accuracy of the statistical static timing analysis method. Next, we demonstrate the delay and power optimization results under the condition that the wire capacitance fluctuates.

The circuits used for the experiments are taken from ISCAS85 and LGSynth93 benchmark sets. These circuits are synthesized and mapped by a commercial logic synthesis tool[10] such that the power dissipation is minimized under some delay constraints. The minimum delay constraint given to each circuit is the reachable minimum delay time of its circuit. The ratio of the total gate capacitance and the total wire capacitance is about 1:1. The target library is a standard cell library used for actual fabrication in a 0.35  $\mu\text{m}$  process with three metal layers. The library includes basic and complex gates. Buffer and Inverter have eleven varieties in the driving strength and other gates have six varieties. A typical delay time at each gate is calculated based on two dimensional look-up tables with capacitive load and slew as parameters. We consider the delay difference between rise/fall transitions. The energy dissipated at each gate, which includes capacitive and short-circuit power dissipation, is derived from a look-up table with capacitive load and slew as parameters. The look-up tables of the gate delay, the transition time of the output signal and the power dissipation are characterized by circuit simulation. As for the power evaluation, we assume that all gates have the same switching probability of 0.2 and the cycle time of the input patterns is 100ns.

## 5.1 Accuracy of Statistical Static Timing Analysis

We verify the accuracy of the statistical static timing analysis. We assume that each gate delay time fluctuates according to normal distribution. The mean is the typical gate delay time and the standard deviation is 20% of its gate delay time. We evaluate the delay time as  $x_1$  in Eq. 10. The evaluated delay time corresponds to mean+3 $\sigma$  in a normal distribution. We compare three methods, Monte Carlo simulation, the statistical static timing analysis with Method1(Section 2.2) which is equivalent to Ref. [4], and the proposed statistical static timing analysis with Method2(Section 2.2). In Monte Carlo simulation, the number of evaluation is 100,000. The comparison of the accuracy is shown in Table 1. The column under "Typ. Delay" is the circuit delay time with no delay fluctuation. The columns "Monte Carlo", "SSTA[4]", "Proposed SSTA" correspond to the result of Monte Carlo simulation, the statistical static timing analysis in Ref. [4] and the proposed statistical static timing analysis respectively. The columns "Delay" are the circuit delay time with delay fluctuation. "Increase" means the proportion of the delay time increase caused by delay fluctuation. "Error" represents the estimation error compared with Monte Carlo simulation. The estimation error range of our method is  $-0.8 \sim 2.9\%$ , and the average error is 1.4%. As for SSTA[4], the range is  $-6.7 \sim -2.7\%$ , and the average is 4.3%. The improvement of the approximation to normal distribution contributes to reduce the estimation error.

## 5.2 Delay and Power Optimization under Wire Capacitance Uncertainties

We demonstrate the delay and power optimization results under wire capacitance uncertainties(Section 4.1). We assume that the wire capacitance fluctuates according to a normal distribution. The mean is the value used in the logic synthesis. The standard deviation is 50% of its mean value, which corresponds to the delay uncertainties of 20% or less.

First, we show the delay optimization results. We optimize the circuits to minimize the delay time. The initial circuits used for this experiment is generated under the constraint of the reachable minimum delay time. Table 2 shows the delay optimization results. "Initial" and "Optimized" correspond to the initial circuit before the optimization and the circuit optimized for delay minimization

respectively. The column "CPU Time" represents the CPU Time for optimization on Alpha Station. Our method reduces the delay time by 8.4% on average. This result shows that the circuit optimized without the consideration of fluctuations is not optimal. The optimization method considering statistical variation is effective for getting better circuits.

Next, we show the power optimization results(Table 3). We optimize the power dissipation under the delay constraints of the initial delay time. Our method reduces power dissipation by 9.3% on average and area by 5.1% without the increase of delay time.

## 6. CONCLUSION

We propose a performance optimization method based on statistical static timing analysis. We develop a technique that improves the accuracy of the worst delay analysis. We devise a new measure that represents the timing criticality at each gate and show the optimization algorithm utilizing the measure. Applications of our optimization method are discussed. The accuracy of statistical static timing analysis is verified experimentally. The maximum estimation error is within 3%. We also demonstrate that our method can reduce delay and power dissipation from the circuits optimized without the consideration of fluctuation. Future work includes implementation and evaluation of the applications discussed in Section 4.

## Acknowledgments

This work is supported in part by Semiconductor Technology Academic Research Center (STARC).

## REFERENCES

- [1] H.-F. Jyu, S. Malik, S. Devadas and K. W. Keutzer, "Statistical Timing Analysis of Combinational Logic Circuits," *IEEE Trans. VLSI Systems*, Vol. 1, No. 2, pp.126-137, June 1993.
- [2] L. Rung-Bin and W. Meng-Chiou, "A New Statistical Approach to Timing Analysis of VLSI Circuits," *Proc. 11th International Conference on VLSI Design*, pp.507-513, 1997.
- [3] R. B. Brashear, N. Menezes, C. Oh, L. T. Pillage and M. R. Mercer, "Predicting Circuit Performance Using Circuit-level Statistical Timing Analysis," *Proc. European Design and Test Conference*, pp.332-337, 1994.
- [4] M. Berkelaar, "Statistical Delay Calculation, a Linear Time Method," *Proc. TAU*, pp.15-24, 1997.
- [5] E.T.A.F. Jacobs and M.R.C.M. Berkelaar, "Gate Sizing Using a Statistical Delay Model," *Proc. DATE2000*, to appear.
- [6] R. B. Hitchcock, G. L. Smith and D. D. Cheng, "Timing Analysis of Computer Hardware," *IBM Journal of Research and Development*, Vol. 26, No. 1, pp.100-105, January 1982.
- [7] O. Coudert, "Gate Sizing for Constrained Delay/Power/Area Optimization," *IEEE Trans. VLSI Systems*, Vol. 5, No. 4, pp. 465-472, December 1997.
- [8] D. -S. Chen and M. Sarrafzadeh, "An Exact Algorithm for Low Power Library-Specific Gate Re-Sizing," *Proc. Design Automation Conference*, pp. 783-788, 1996.
- [9] M. Hashimoto, H. Onodera, and K. Tamaru, "A Power and Delay Optimization Method using Input Reordering in Cell-Based CMOS Circuits," *IEICE Trans. Fundamentals*, Vol. E82-A, No. 1, pp. 159-166, January 1999.
- [10] Synopsys Inc., Design Compiler Reference Manual, 1999.

Table 1: Accuracy of Statistical Static Timing Analysis

Circuit	Typ. Delay (ns)	Monte Carlo		SSTA[4]		Proposed SSTA	
		Delay (ns)	Increase (%)	Delay (ns)	Error (%)	Delay (ns)	Error (%)
C432_A	4.48	5.57	24.3	5.39	-3.2	5.65	1.3
C432_B	4.97	6.10	22.7	5.90	-3.3	6.19	1.5
C432_C	5.91	7.13	20.6	6.94	-2.7	7.26	1.8
C432_D	6.92	8.58	24.0	8.35	-2.7	8.79	2.4
C3540_A	6.71	8.28	23.4	7.97	-3.7	8.43	1.8
C3540_B	7.18	8.77	22.1	8.45	-3.6	8.95	2.1
C3540_C	7.97	9.65	21.1	9.30	-3.6	9.80	1.6
C3540_D	8.92	10.69	19.8	10.32	-3.5	10.90	2.0
C5315_A	6.00	7.73	28.8	7.31	-5.4	7.83	1.3
C5315_B	6.97	8.58	23.1	8.26	-3.7	8.74	1.9
C5315_C	7.98	9.74	22.1	9.48	-2.7	10.02	2.9
C5315_D	8.90	10.77	21.0	10.47	-2.8	11.03	2.4
C7552_A	4.84	6.12	26.4	5.86	-4.2	6.20	1.3
C7552_B	5.02	6.28	25.1	5.98	-4.8	6.33	0.8
C7552_C	5.99	7.39	23.4	7.07	-4.3	7.48	1.2
C7552_D	6.95	8.53	22.7	8.18	-4.1	8.68	1.8
alu4_A	3.31	4.25	28.4	4.00	-5.9	4.23	-0.5
alu4_B	3.99	5.10	27.8	4.76	-6.7	5.10	0.0
alu4_C	4.95	6.18	24.8	5.82	-5.8	6.14	-0.6
alu4_D	5.83	7.26	24.5	6.80	-6.3	7.20	-0.8
des_A	3.60	4.73	31.4	4.52	-4.4	4.78	1.1
des_B	3.98	5.26	32.2	5.00	-4.9	5.26	0.0
des_C	4.96	6.50	31.0	6.12	-5.8	6.46	-0.6
des_D	5.91	7.52	27.2	7.17	-4.7	7.59	0.9
Average	-	-	24.9	-	4.3	-	1.4

Table 2: Delay Optimization

Circuit	Initial			Optimized				CPU Time (s)	#Gates
	Delay (ns)	Area (mm <sup>2</sup> )	Power (mW)	Delay (ns)	Delay Reduction(%)	Area (mm <sup>2</sup> )	Power (mW)		
C432_A	5.22	0.017	33	4.86	6.9	0.018	34	12	178
C3540_A	7.60	0.083	147	7.00	7.9	0.088	159	462	871
C5315_A	7.17	0.089	138	6.39	10.9	0.093	147	260	1001
C7552_A	5.58	0.134	234	5.19	7.0	0.138	243	695	1339
alu4_A	3.96	0.122	244	3.65	7.8	0.126	254	224	1386
des_A	4.56	0.214	383	4.11	9.9	0.214	389	2836	2252
Average	-	-	-	-	8.4	-	-	-	-

Table 3: Power Optimization

Circuit	Initial			Optimized				CPU Time (s)
	Delay (ns)	Area (mm <sup>2</sup> )	Power (mW)	Area (mm <sup>2</sup> )	Area Reduction(%)	Power (mW)	Power Reduction(%)	
C432_A	5.22	0.017	33	0.016	5.9	29	12.1	3
C3540_A	7.60	0.083	147	0.079	4.8	135	8.2	100
C5315_A	7.17	0.089	138	0.087	2.2	131	5.1	79
C7552_A	5.58	0.134	234	0.126	6.0	209	10.7	409
alu4_A	3.96	0.122	244	0.116	4.9	220	9.8	290
des_A	4.56	0.214	383	0.199	7.0	346	9.7	5447
Average	-	-	-	-	5.1	-	9.3	-