PAPER Special Section on VLSI Design and CAD Algorithms

A Performance Optimization Method by Gate Resizing Based on Statistical Static Timing Analysis

Masanori HASHIMOTO[†], ***Student Member and*** **Hidetoshi ONODERA**[†], ***Regular Member***

SUMMARY This paper discusses a gate resizing method for performance enhancement based on statistical static timing analysis. The proposed method focuses on timing uncertainties caused by local random fluctuation. Our method aims to remove both over-design and under-design of a circuit, and realize high-performance and high-reliability LSI design. The effectiveness of our method is examined by 6 benchmark circuits. We verify that our method can reduce the delay time further from the circuits optimized for minimizing the delay without the consideration of delay fluctuation.

key words: *statistical static timing analysis, static timing analysis, gate resizing, transistor sizing, performance optimization*

1. Introduction

There are several sources that cause the uncertainties of circuit delay time, such as manufacturing fluctuation, estimation error of wire capacitance and resistance, uncertainties of wire capacitance during physical design, supply voltage and temperature change, 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, mean+3σ. 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 confidence 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 Calro simulation-based techniques, so these methods are not suitable for performance optimization method from the point of computation time. The method proposed by Berkerlaar in Refs. [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 (corresponding to mean+3σ, for example) [4], because of the definition of the worst-case delay and the approximation method used in Ref. [4]. In a statistical analysis, it is important to estimate a statistically well-defined worst-case value. We therefore define the worst-case delay in a statistical manner, and device a technique to improve the accuracy of the worst-case delay estimation. This method is utilized 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 constrains are expressed as analytic forms. In this method, the delay should be represented by 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

Manuscript received March 31, 2000.
Manuscript revised June 23, 2000.

[†]The authors are with the Department of Communications and Computer Engineering, Kyoto University, Kyoto-shi, 606-8501 Japan.

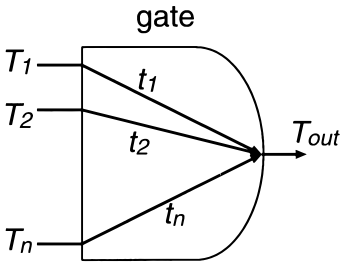

Fig. 1 Gate delay model.

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, Sect. 6 concludes the discussion.

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 approximation methods 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 Sect. 2, we do not distinguish rise/fall transitions for simplifying the explanation. But the real implementation in Sect. 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.

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

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

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 explain this method briefly. Next, we define the worst-case delay of the circuit with delay fluctuation and discuss a technique that improves the accuracy of the worst-case delay calculation.

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 μ_{T_i} and standard deviation σ_{T_i}. We also assume that the gate delay time from the i-th input to the output is distributed normally with a stochastic variable t, mean μ_{t_i} and standard deviation σ_{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 distribution of f_i becomes a normal distribution $\mathrm{N}(\mu_{T_i}+\mu_{t_i}, \sqrt{\sigma_{T_i}^2+\sigma_{t_i}^2})$. The cumulative distribution function F_i is defined as follows.

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

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.

$$P(C \le x) = P((A \le x) \cap (B \le x)), \tag{3}$$

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.

$$P(C \le x) = P(A \le x) \cdot P(B \le x). \tag{4}$$

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

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

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

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

Equation (6) can be rewritten as follows.

$$\begin{aligned} P(C = x) \;=\; & P(A = x)\cdot P(B \le x) \\ & + P(B = x)\cdot P(A \le x). \end{aligned} \tag{7}$$

Extending Eq. (6) for n stochastic variables, the probability density function f_{out}, which corresponds to

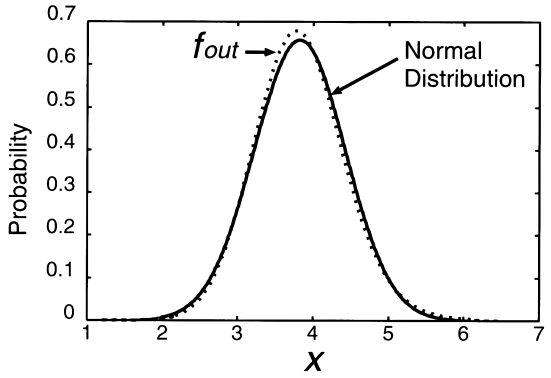

Fig. 2 Difference between f_{out} and a normal distribution.

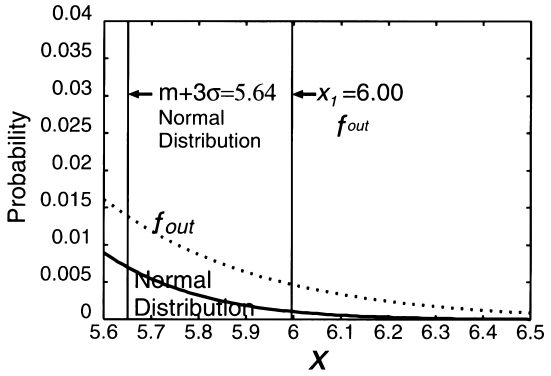

Fig. 3 Difference between f_{out} and a normal distribution (magnified).

the distribution of the latest arrival time T_{out}, can be represented as follows.

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

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.

We discuss the definition of the worst-case delay under the statistical delay model. The distribution of the latest arrival time, f_{out}, is different from a normal distribution, though assumed to be normal. Figures 2 and 3 show an example of the difference between f_{out} and the normal distribution. The function 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 3, 1, 3.6, 0.6 and 2 respectively. We calculate the mean m and standard deviation σ of f_{out} according to the definition, and generate the normal distribution $\mathrm{N}(m, \sigma)$. If the distribution of f_{out} is exactly normal, x_1 in the following equation becomes equal to $m + 3\sigma$.

$$0.9986501 = \int_{-\infty}^{x_1} f_{out}(x)dx, \quad (9)$$

where the value 0.9986501 is the probability of a normal distribution between $-\infty$ and $m + 3\sigma$. But in reality, x_1 of f_{out} is different from $m + 3\sigma$. The value x_1 is 6.00, whereas $m + 3\sigma$ is 5.64. This difference derives from the fact that the curve of f_{out} falls slower than it rises. If the worst-case delay is defined as $m + 3\sigma$, the lower probability of $x \leq m + 3\sigma$ becomes smaller than 99.87%. The actual value of the lower probability varies depending on the shapes of f_1 and f_2. On the other hand, when the worst-case delay is defined as x_1, the lower probability of $x \leq x_1$ becomes a fixed value of 99.87%. In statistical analysis, evaluating the delay time with a fixed lower probability is important. We therefore define the worst-case delay as x_1 in Eq. (9). When we evaluate the delay with the different lower probability, the value of the left term in Eq. (9) should be changed accordingly. Hereafter, the worst-case delay is defined as x_1 in Eq. (9).

Next, we discuss the approximation of f_{out} to a normal distribution. In Ref. [4], f_{out} is approximated as a normal distribution to reduce computational costs. Our method also approximate f_{out} to a normal distribution. Here, we examine the approximation methods of f_{out} from the viewpoint whether the worst-case delay x_1 can be calculated accurately. Equation (9) is rewritten using Eq. (8) as follows.

$$0.0013499 = \int_{x_1}^{\infty} \sum_{i}^{n} \left[f_i(x) \cdot \prod_{j \neq i}^{n} F_j(x) \right] dx. \quad (10)$$

The value x_1 of each f_i is close to or smaller than x_1 of f_{out}. In the range of x between x_1 and ∞, the cumulative distribution function $F_j(x)$ is almost 1. In order to calculate the worst-case delay x_1 accurately, the approximation accuracy of f_i where x is larger than x_1 is important. We therefore should approximate f_{out} well in the region where x is close to and larger than x_1 of f_{out}, which contributes the accurate calculation of x_1 at the fan-out gates that the gate drives. We compare two approximation method of f_{out} to a normal distribution $\mathrm{N}(m, \sigma)$.

Method 1 Calculate the mean m and the standard deviation σ of f_{out} according to the definition.

Method 2 Find the values of x_0 and x_1 that satisfy Eqs. (9) and (11). The mean m is calculated as $(x_0 + x_1)/2$ and the standard deviation σ is $(x_1 - x_0)/6$.

$$0.0013499 = \int_{-\infty}^{x_0} f_{out}(x)dx. \quad (11)$$

Method 1 is adopted in Ref. [4]. In Method 2, a value x_0 corresponds to $m - 3\sigma$ of a normal distribution and x_1 to $m + 3\sigma$ from the viewpoint of the lower and upper probability. Method 2 adjusts x_1 of the approximated normal distribution to x_1 of f_{out}. Figure 4 shows the approximation results of Method 1 and Method 2.

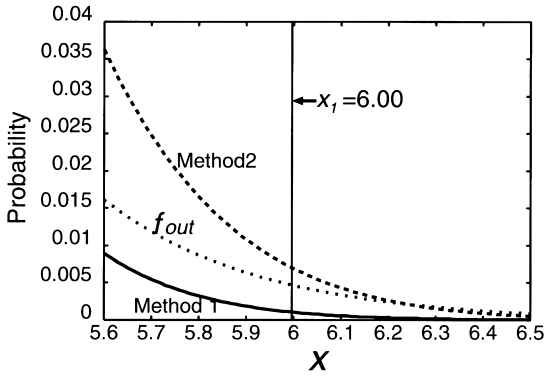

Fig. 4 Approximation to normal distribution (magnified).

Method 1 underestimates the delay time. On the other hand, in Method 2, the distribution shape of f_{out} where x is larger than x_1 is well approximated. Therefore, Method 2 is suitable for the approximation to calculate the worst-case delay x_1 accurately. When the definition of the worst-case delay is changed, i.e. the value of the left term in Eq. (9) becomes other value, Method 2 is modified as follows. For example, suppose the value of the left term in Eq. (9) becomes 0.97725, which corresponds to the probability of $x \leq m + 2\sigma$ in a normal distribution. The value of left term in Eq. (11) becomes 0.02275. The standard deviation σ is calculates as $(x_1 - x_0)/4$.

The discussion so far assumes that the distribution of gate delay is normal and hence the probability density function f_i is a normal distribution. In this case, the probability density function f_{out}, although it is not a normal distribution, can be approximated to a normal distribution. We showed two methods for the approximation. Please notice that the essence of the statistical static timing analysis explained from Eqs. (2) through (10) does not require that f_i is normal. Thus, if the probability density function f_i is not normal, we can still apply Eq. (8) to calculate the probability density function f_{out}. In this case, we need to find an appropriate function for f_i and f_{out} or need to calculate f_i and f_{out} numerically. In any case, through the successive calculation of the probability density function from the primary input to the primary output, we can perform statistical static timing analysis.

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 performance 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 σ 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.

Before the detailed explanation of "criticality," we explain the concept of "criticality." Under the statistical delay model, many paths have a possibility to become the longest path. In other words, many gates have an effect to the distribution of the total circuit delay. To speak more rigidly, all gates have an influence to the circuit delay distribution although the magnitude of the influences is different. Therefore, we should define the timing criticality at each gate as the magnitude of the statistical influence to the circuit delay distribution. Namely, the gate that has a strong statistical influence to the total delay distribution should be defined as critical. We then model the statistical impact of each gate delay to the total circuit delay as the measure of timing criticality named "criticality," using a heuristic numerical expression. In this model, large "criticality" represents high timing criticality, thus the gate with large "criticality" should be resized for reducing the circuit delay. When "criticality" is zero, the gate has no statistical influence to the circuit delay distribution. So, the gate with small "criticality" could be downsized for reducing power dissipation without delay increase. Given the measure of "criticality," our method can choose a candidate of gate resizing efficiently. Hereafter, we explain the details of "criticality."

The term in the bracket of Eq. (8) represents the following probability.

$$f_i(x) \cdot \prod_{j \neq i}^{n} F_j(x) = P(T_i + t_i = x) \cdot \prod_{j \neq i}^{n} P(T_j + t_j \leq x). \quad (12)$$

The input with the high probability of Eq. (12) affects the distribution of T_{out} at x strongly. The probability of Eq. (12) 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.

$$influence_i = C_1 \cdot \int_{x_1}^{\infty} f_i(x) \cdot \prod_{j \neq i}^{n} F_j(x) \cdot \exp(C_2 \cdot x) dx, \quad (13)$$

where C_1 is a normalization coefficient to satisfy $\sum_i^n influence_i = 1$ and C_2 is a constant. A term

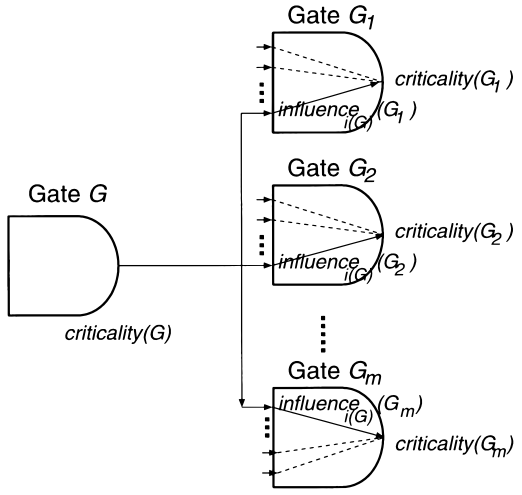

Fig. 5 Propagation of "criticality."

$\exp(C_2 \cdot x)$ is multiplied in order to emphasize the region of large arrival time. However, this is not a primary term for the definition of $influence_i$. Also, according to our experiments, the value of C_2 is not so sensitive to $influence_i$. We empirically decide the value of C_2 such that the value $\exp(C_2 \cdot x)$ increases by 50% when time x increases by 0.1 ns around the time of our interest. 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. 5 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 $influence_{i(G)}(G_j) \cdot criticality(G_j)$.

$$criticality(G) = \sum_{j}^{m} influence_{i(G)}(G_j) \cdot criticality(G_j), \quad (14)$$

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. (14) 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.

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 to the size with minimum delay.
- Step 5: if there are no sizes to reduce delay, 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. We change the size of the gate four times, i.e. 2 size-up, 1 size-up, 1 size-down, and 2 size-down, and evaluate the circuit delay for each case. We choose the size that the circuit delay decrease the most, and resize the gate to the size. If the resizing does not decrease the circuit delay, we restore the gate size and remove the gate from L and go back to Step 2. Otherwise, we go back to Step 1. Our algorithm searches a solution greedily, so our algorithm necessarily reaches the condition that the circuit delay does not decrease by resizing the gates in the circuit. In this condition, as the steps between Step 2 and Step 5 are repeated, the number of the elements in the list L decreases. Finally the list L becomes empty and the optimization procedure finishes in Step 2.

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 to the size with minimum power dissipation without delay violation.
Step 5: if there are no gate sizes to choose, 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 influences on the circuit delay. The size of the found gate is changed to 2 size-down and 1 size-down from the initial size, and we evaluate the circuit delay and the power dissipation for each case. The found gate is down-sized to the size that makes the power dissipation minimum without the delay violation. If the resizing does not reduce power dissipation without delay violation, we remove the gate from L and go back to Step 2. Otherwise, we go back to Step 1. At the end of the optimization, there become no gates to reduce power dissipation without delay violation. The list L becomes empty by the repetitions between Step 2 and Step 5, and finally the optimization procedure finishes.

The optimization algorithm explained above has the possibility of falling into a bad local minimum solution. In order to escape from a bad local minimum solution, we optimize the circuit delay a little bit, such as 0.1% of its circuit delay, using the algorithm in Sect. 3.1. 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. We assume that the gate delay fluctuation discussed in this section can be approximated to a normal distribution. If the distribution is not normal, we can still perform statistical timing analysis as described in Sect. 2.2. In this case, we need to modify the method for expressing the probability density functions.

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.

$$delay = a_0 + a_1 \cdot t_{tran} + a_2 \cdot c_{load} + a_3 \cdot t_{tran} \cdot c_{load}, \quad (15)$$

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_t, β, ...) and the process parameters (t_{ox}, W, L, ...). 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.

$$\delta delay = \sum_i d_i \cdot \delta p_i, \quad (16)$$

where d_i is a constant. In the case of the local fluctuation, δp_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 worst-case delay estimation. Next, we demonstrate the delay fluctuation caused by the timing uncertainties of local random

Table 1 Accuracy of worst-case delay calculation.

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
	-	-	24.9	-	4.3	-	1.4

change. We finally show 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] under a reasonable wire load model such that the power dissipation is minimized under the following four delay constraints. The circuits labeled "_A" are generated under the minimum as well as reachable delay constraints of the respective circuits. The delay constraints given to the circuit with "_B," "_C" and "_D" are made loose gradually in this order. 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 μm 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 100 ns.

5.1 Accuracy of Worst-Case Delay Calculation

We verify the accuracy of the worst-case delay calculation. 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 worst-case delay time defined as x_1 in Eq. (9). We compare three methods, Monte Carlo simulation, the statistical static timing analysis with Method1 (Sect. 2.2) which is equivalent to Ref. [4], and the proposed statistical static timing analysis with Method2 (Sect. 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 results 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 worst-case delay time of the circuits with delay fluctuation. "Increase" means the proportion of the difference between the typical (no fluctuation) delay and the worst-case delay with delay fluctuation. "Error" represents the estimation error compared with Monte Carlo simulation. The range of the estimation error in 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 a better calculation of the worst-case delay x_1.

5.2 Circuit Delay Fluctuation—Case Study—

We demonstrate the circuit delay fluctuation caused by the timing uncertainties of local random fluctuation. We first discuss the delay uncertainty sources, and make an assumption of the delay uncertainty sources. We then show the result of the statistical static timing analysis under this assumption.

5.2.1 Assumption of Delay Fluctuation Sources

As for the sources of delay fluctuation, we take up two sources; manufacturing variability and design uncertainties of wire capacitance.

Manufacturing Variability

The manufacturing variability consists of two components; the variability in transistor characteristics and the variability in interconnect structure. We first discuss the transistor characteristics. The fluctuation is composed of local components (different for individual gates in a circuit) and global components (the same for all gates in a circuit) [12]. In the process used for the experiments, the worst-case delay evaluated from the given worst-case SPICE parameters is 30% larger than the typical-case delay. Thus, if we assume that the ratio of the local fluctuation component and the global fluctuation component is 2:1, 3σ of the local delay variability becomes 20%.

We next examine the variability in interconnect structure. Reference [11] analyzes the decomposition of the delay variability due to manufacturing fluctuation. The analysis indicates that the interconnect is responsible for 12 to 18% of the total delay variability and the rest (82 to 88%) is contributed by transistors. With this ratio of each contribution, 3σ of the total delay variability becomes 24%. Thus, in this case study, the standard deviation of the delay due to transistor and interconnect variabilities is estimated to be 8%.

Design Uncertainties of Wire Capacitance

The estimated wire capacitances during layout design are different from the capacitances of the final layout. At cell placement design phase, there are uncertainties in wire route and adjacencies. Recently the proportion of the coupling capacitance between adjacent wires increases, which results in the increase of uncertainties at placement phase. We evaluate the ratio of the estimated capacitance at placement phase compared with the capacitance of the final layout using a 32-bit CPU circuit (about 13k cells). Figure 6 shows the distribution of the estimation error of the wire capacitance at cell placement phase. Even when the cell place is fixed, there is the wire capacitance uncertainty of which the standard deviation is 25% of the estimated capacitance.

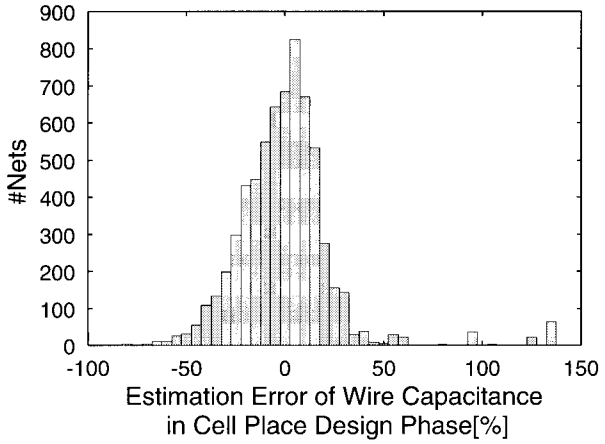

Fig. 6 Distribution of wire capacitance uncertainties at cell placement design phase.

RC extraction tools have a certain amount of estimation errors. The amount of errors in capacitance extraction may vary depending on the used algorithm (2D, quasi 3-D, 3D etc.) as well as on the complexity of the interconnect structures under extraction. It is not easy to estimate the uncertainty in the extraction, but we think that the standard deviation of 10% may be a reasonable guess.

Summary of Uncertainties

From the above discussion, the assumption of the delay uncertainty sources is summarized as follows.

Manufacturing Variability The delay time of each gate fluctuates such that the mean delay is its typical delay time and the standard deviation is 8% of its typical delay.

Extraction Error The extracted wire capacitance has the error of which σ is 10% of the extracted value.

Uncertainty at Placement The wire capacitance estimated at cell placement design phase has the uncertainty of wire capacitance. The mean is the estimated value and the standard deviation is 25% of the estimated value.

5.2.2 Results

We evaluate the worst-case delay time as x_1 in Eq. (9), which corresponds to mean+3σ in a normal distribution, under each uncertainty source. The result of statistical timing analysis is shown in Table 2. The column under "Typ. Delay" is the circuit delay time

Table 2 Delay fluctuation.

Circuit	Typ. Delay (ns)	Manufacturing Variability		Extraction Error		Uncertainty at Placement		MV+EE		MV+EE +UP		CPU Time (s)	#Gates
		Delay (ns)	Inc. (%)	Delay (ns)	Inc. (%)	Delay (ns)	Inc. (%)	Delay (ns)	Inc. (%)	Delay (ns)	Inc. (%)		
C432_A	4.48	4.89	9.2	4.52	0.9	4.76	6.3	4.90	9.4	4.98	11.1	0.03	178
C432_B	4.97	5.39	8.5	5.02	1.0	5.19	4.4	5.40	8.7	5.49	10.5	0.03	154
C432_C	5.91	6.37	7.8	6.00	1.5	6.21	5.1	6.40	8.3	6.52	10.3	0.03	144
C432_D	6.92	7.60	9.8	7.08	2.3	7.41	7.1	7.63	10.3	7.82	13.0	0.03	130
C3540_A	6.71	7.32	9.1	6.77	0.9	7.04	4.9	7.32	9.1	7.39	10.1	0.17	871
C3540_B	7.18	7.78	8.4	7.25	1.0	7.51	4.6	7.79	8.5	7.89	9.9	0.16	835
C3540_C	7.97	8.61	8.0	8.13	2.0	8.56	7.4	8.63	8.3	8.94	12.2	0.16	703
C3540_D	8.92	9.59	7.5	9.06	1.6	9.46	6.1	9.62	7.8	9.86	10.5	0.16	657
C5315_A	6.00	6.60	10.0	6.13	2.2	6.44	7.3	6.62	10.3	6.82	13.7	0.28	1001
C5315_B	6.97	7.61	9.2	7.15	2.6	7.54	8.2	7.65	9.8	7.89	13.2	0.25	946
C5315_C	7.98	8.69	8.9	8.17	2.4	8.59	7.6	8.73	9.4	9.01	12.9	0.25	932
C5315_D	8.90	9.65	8.4	9.12	2.5	9.62	8.1	9.70	9.0	10.04	12.8	0.26	919
C7552_A	4.84	5.33	10.1	4.93	1.9	5.16	6.6	5.34	10.3	5.47	13.0	0.29	1339
C7552_B	5.02	5.49	9.4	5.11	1.8	5.34	6.4	5.51	9.8	5.63	12.2	0.29	1248
C7552_C	5.99	6.49	8.3	6.08	1.5	6.43	7.3	6.52	8.8	6.72	12.2	0.31	1127
C7552_D	6.95	7.56	8.8	7.12	2.4	7.52	8.2	7.61	9.5	7.86	13.1	0.32	1087
alu4_A	3.31	3.63	9.7	3.37	1.8	3.58	8.2	3.64	10.0	3.74	13.0	0.24	1386
alu4_B	3.99	4.40	10.3	4.11	3.0	4.38	9.8	4.43	11.0	4.61	15.5	0.26	1219
alu4_C	4.95	5.35	8.1	5.10	3.0	5.46	10.3	5.40	9.1	5.67	14.5	0.31	1184
alu4_D	5.83	6.30	8.1	6.06	3.9	6.50	11.5	6.37	9.3	6.72	15.3	0.34	1167
des_A	3.60	4.02	11.7	3.70	2.8	3.98	10.6	4.04	12.2	4.20	16.7	1.00	2252
des_B	3.98	4.44	10.6	4.16	4.5	4.51	13.3	4.47	12.3	4.75	19.3	1.26	1927
des_C	4.96	5.50	10.9	5.23	5.4	5.69	14.7	5.58	12.5	5.94	19.8	1.25	1769
des_D	5.91	6.49	9.8	6.14	3.9	6.62	12.0	6.55	10.8	6.93	17.3	0.87	1714
Average	-	-	9.2	-	2.4	-	8.2	-	9.8	-	13.4	-	-

with no delay fluctuation. The columns "Manufacturing Variability," "Extraction Error" and "Uncertainty at Placement" correspond to the results under each uncertainty source respectively. The columns "Delay" are the worst-case circuit delay time with delay fluctuation. "Inc." means the percentage of the delay time increase caused by delay fluctuation. "MV+EE" is the result under both manufacturing variability and extraction error. This situation corresponds to the final delay evaluation of the completed circuit using an accurate RC extraction tool. "MV+EE+UP" means the situation that the circuit delay is estimated at cell placement design phase. So, the result under all three fluctuation sources is listed below "MV+EE+UP." The column "CPU Time" represents the CPU Time for timing analysis on Alpha Station.

Due to manufacturing variability, extraction error, and uncertainty at placement, the worst-case circuit delay increases by 9.2%, 2.4% and 8.2% on average from the delay without fluctuation, respectively. The amount of increase varies from circuit to circuit under the same uncertainty sources. For example, the increase caused by the uncertainty at placement ranges from 4.4% to 14.7%, which indicates that the impact of uncertainty is considerably different in each circuit.

In the evaluation of the circuit from the final layout ("MV+EE"), the delay increases by 9.8% on average from the typical delay. This result indicates that the circuit design does not succeed without the consideration of local delay uncertainties. In the case of the delay estimation at cell placement design phase ("MV+EE+UP"), there is a possibility that the delay time increases by 13.4%.

5.3 Delay and Power Optimization under Wire Capacitance Uncertainties

We demonstrate the delay and power optimization results under wire capacitance uncertainties. 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. Please note that the initial circuits used for this experiment are synthesized and optimized for minimizing the circuit delay under the deterministic delay model. Table 3 shows the delay optimization results. "Initial" and "Optimized" correspond to the initial circuit before the optimization and the circuit optimized for delay minimization respectively. "Area" is calculated as the sum of the cell area. 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.

Table 3 Delay optimization.

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

Table 4 Power optimization.

Circuit	Initial			Optimized				CPU
	Delay (ns)	Area (mm^2)	Power (mW)	Area (mm^2)	Area Reduction(%)	Power (mW)	Power Reduction(%)	Time (s)
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	-

Next, we show the power optimization results (Table 4). 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-case delay analysis. We device a new measure that represents the timing criticality at each gate and show the optimization algorithm utilizing the measure. The accuracy of the worst-case delay calculation is verified experimentally. The maximum estimation error is within 3%. We evaluate the delay fluctuation under some of the delay uncertainty sources. We also demonstrate that our method can reduce delay and power dissipation from the circuits optimized without the consideration of fluctuation.

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] R.-B. Lin and M.-C. Wu, "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, pp.283–290, 2000.
[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, Jan. 1982.
[7] O. Coudert, "Gate sizing for constrained delay/power/area optimization," IEEE Trans. VLSI Systems, vol.5, no.4, pp.465–472, Dec. 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, Jan. 1999.
[10] Synopsys Inc., Desigin Compiler Reference Manual, 1999.
[11] M. Orshansky, C. Spanos, and C. Hu, "Circuit performance variability decomposition," Proc. Int'l Workshop on Statistical Metrology, pp.10–13, 1999.
[12] M. Berkelaar and E. Jacobs, "Sources and quantification of delay variations in a 250 nm CMOS digital cell library," Proc. IWLS, pp.335–339, 2000.

Masanori Hashimoto received the B.E. and M.E. degrees in electronic engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, respectively . Currently, he is studying toward the Ph.D. degree in the Department of Communications and Computer Engineering, Kyoto University. He is a Research Fellow of the Japan Society for the Promotion of Science. He received the IEEE Solid-State Circuit Society Chapter's Award in 1999. His research interest includes computer-aided-design for digital integrated circuits. He is a member of IEEE and IPSJ.

Hidetoshi Onodera received the B.E., and M.E., and Dr.Eng. degrees in Electronic Engineering from Kyoto University, Kyoto, Japan, in 1978, 1980, 1984, respectively. Since 1983 he has been an Instructor (1983–1991), an Associate Professor (1992–1998), a Professor (1999-) in the Department of Communications and Computer Engineering, Graduate School of Infomatics, Kyoto University. His research interests include computer-aided-design for integrated circuits, and analog and mixed analog-digital circuits design. He is a member of the Information Processing Society of Japan, and IEEE.