A Statistical Static Timing Analysis Considering Correlations Between Delays

Shuji Tsukiyama[+] Masakazu Tanaka[*] Masahiro Fukui[*]

+ Dept. of EECE
Chuo University
Tokyo, JAPAN 112-8551
tsuki@elect.chuo-u.ac.jp

* Advanced LSI Tech. Development Center
Matsushita Electric Industrial Co., Ltd.
Nagaokakyo, JAPAN 617-8520
{tanack, fukui}@ngk.csdd.mei.co.jp

Abstract: In this paper, we present a new algorithm for the statistical static timing analysis of a CMOS combinatorial circuit, which can treat correlations of arrival times of input signals to a logic gate and correlations of switching delays in a logic gate. We model each switching delay by a normal distribution, and use a normal distribution of two stochastic variables with a coefficient of correlation for computing the distribution of output delay of a logic gate. Since the algorithm takes the correlation into account, the time complexity is O(n•m) in the worst-case, where n and m are the numbers of vertices and edges of the acyclic graph representing a given combinatorial circuit.

I. INTRODUCTION

The importance of statistical static timing analysis[1-5] is increasing in designing high density, high speed and low power VLSIs in deep sub-micron era. Because, designers often set excessive margins derived from the worst-case analysis in order to avoid the effect of the delay time uncertainty, and such excessive margins usually bring over-design of circuits. If designers can estimate the distribution of critical path delay caused by all local uncertainties, over-design of circuits may be eliminated so that high density and high performance VLSIs are produced with high yield[6-8].

Although several researches have been done on this topic[1-7], all of them except [5] assume that the distributions of all signal delays are independent each other. However, such an assumption is not realistic in combinatorial circuits with re-convergent paths. For example, if the delays of signals x and y of the circuit shown in Fig. 1 are heavily depend on the delay of signal v, then we cannot ignore the correlation between delays of signals x and y in computing the maximum delay of signal z.

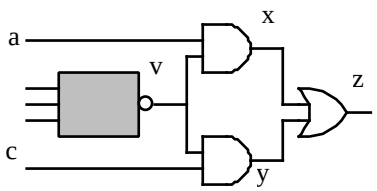

Fig. 1 Arrival times of signals x and y have a correlation

In this paper, we present a new algorithm for the statistical static timing analysis of a CMOS combinatorial circuit, which can treat not only correlations of delays of signals denoted in Fig. 1 but also correlations between delays of transistors contained in a logic gate. This algorithm assumes that the delay of each logic gate is modeled by a normal distribution (Gaussian distribution), as usually done in the previous researches. In order to compute the distribution of the output delay of a logic gate, a maximum operation on the stochastic variables is necessary, and the proposed algorithm uses a normal distribution of two stochastic variables with a coefficient of correlation[9] for the maximum operation.

If delays of all signals are independent, the distribution of the maximum delay can be computed in O(n+m) time[3], where n and m are the numbers of vertices and edges of the graph representing a given combinatorial circuit. But, since the proposed algorithm take the correlation into account, the time complexity of the algorithm becomes O(n•m) in the worst-case. However, for real combinatorial circuits, we can expect that the time complexity is much less than O(n•m).

II. PRELIMINARIES

In order to find the distribution of the maximum delay of a CMOS combinatorial circuit, we represent the circuit by an acyclic graph G=(V,E), as shown in Fig. 2. In the figure, each box denotes a logic gate, which is drawn to show the correspondence between the circuit and the graph. Each vertex contained in a box represents a terminal of the corresponding logic gate, and a vertex in the left or right side in a box corresponds to an input or an output terminal, respectively. Each sink, from which no edge goes out, corresponds to a primary output, and T denotes the set of

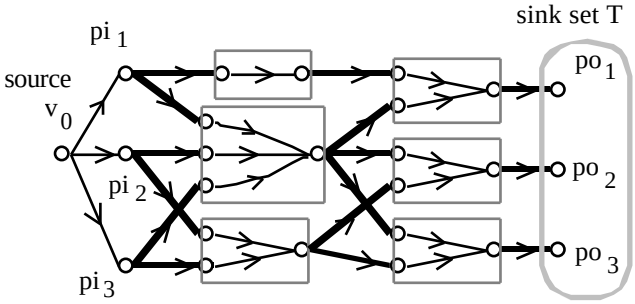

Fig. 2 Acyclic graph G=(V,E) representing a circuit

sinks. Vertex v_0 is the unique source into which no edge comes, and each edge going out from v_0 comes into a vertex corresponding to a primary input. These edges are used to introduce the differences and distributions of arrival times of primary inputs. Each edge in a box goes out from the vertex representing an input of the corresponding logic gate, and comes into the vertex representing the output of the gate. Edges exclusive of the ones going out from v_0 represent interconnects, if they are not contained in a box.

Each edge e=(v,w) contained in a box has delays $t_0(e)$ and $t_1(e)$, such that $t_0(e)$ and $t_1(e)$ denote the switching times of pMOS and nMOS transistors, respectively. Henceforth, we use "b" to indicate 0 or 1, and $t_b(e)$ denotes $t_0(e)$ or $t_1(e)$. Delay $t_b(e)$ has a certain uncertainty, and is a stochastic variable. It is determined by saturated current I_{dsat}, load capacitance C_{load} of the transistor, and slew rate t_{slew} of the gate voltage. The uncertainty of the delay comes from the distribution of I_{dsat}, which mostly depends on the distribution of gate length L_g of the transistor. Since the distribution of L_g can be modeled by a normal distribution like the distribution of threshold voltage V_t [10], we model the distribution of delay $t_b(e)$ of edge e in a box by a normal distribution $N(\mu_b(e), \sigma^2_b(e))$, where $\mu_b(e)$ and $\sigma^2_b(e)$ are the mean and the variance of the distribution of $t_b(e)$.

The distribution of L_g depends on the distributions of space S_{poly} between adjacent polysilicon gates, and gate width W_g and length L_{diff} of diffusion area of the transistor. Therefore, by extracting these quantities from the mask pattern, the distribution of L_g can be estimated, and hence the distribution of the switching delay $t_b(e)$ of a transistor. With the use of the distribution data and the proposed algorithm, we can execute post-layout timing analysis, gate size optimization[6,7], and layout optimization of a macrocell[8].

Since the distribution of L_g depends on the distribution of space S_{poly} between adjacent gates, delay $t_b(e)$ has a correlation with the delay of the edge corresponding to the adjacent transistor. Therefore, we introduce a correlation between delays $t_b(e')$ and $t_b(e'')$ of edges e' and e'' which correspond to transistors with the same type in a logic gate, and denote the coefficient of correlation between these delays by $\rho_b(e',e'')$. Such correlations are introduced only for edges contained in a single logic gate.

For an edge e representing an interconnect, delay $t_b(e)$ designates the interconnect delay, which is assumed to be constant, that is, $\sigma_b(e)=0$. This delay is evaluated by a method such as PRIMO[11]. If interconnect delay is also distributed by a normal distribution, then we may introduce a variance $\sigma_b(e)\neq 0$. But, in such a case, the proposed algorithm must be modified, because the distributions of $t_0(e)$ and $t_1(e)$ may have a correlation.

Now, let us show that the maximum delay to a vertex w in G can be estimated by using the edge-delays introduced above.

Let d(v,0) and d(v,1) be the maximum delays spent for transmitting signal 0 and 1 from source v_0 to a vertex v V, respectively. These are stochastic variables. In the following, the mean and variance of d(v,b) are denoted by $m_b(v) = Exp[\ d(v,b)\ ]$ and $s_b(v) = Var[\ d(v,b)\ ]$, respectively.

Firstly, we consider the case where vertex w corresponds to a sink or an input node of a gate. In this case, only one edge e = (v,w) corresponding to an interconnect comes into w, and hence d(w,0) and d(w,1) can be calculated by the following equations:

$$d(w,0) = d(v,0) + t_0(e), \qquad d(w,1) = d(v,1) + t_1(e).$$

Since $t_0(e)$ and $t_1(e)$ are constant, $m_b(w)$ and $s_b(w)$ are obtained as follows:

$$m_0(w) = m_0(v) + t_0(e), \qquad m_1(w) = m_1(v) + t_1(e),$$
$$s_0(w) = s_0(v), \qquad s_1(w) = s_1(v).$$

Moreover, correlation coefficient $r_{bb'}(w,u) = R[\ d(w,b), d(u,b')\ ]$ between d(w,b) and d(u,b') of a vertex $u \neq w$ can be obtained by the following equations. Henceforth, we use b' together with b to indicate 0 or 1, and b' is not necessarily equal to b.

$$r_{00}(w,u) = r_{00}(v,u), \qquad r_{01}(w,u) = r_{01}(v,u),$$
$$r_{10}(w,u) = r_{10}(v,u), \qquad r_{11}(w,u) = r_{11}(v,u).$$

Obviously, we have

$$r_{00}(w,w) = r_{11}(w,w) = 1,$$
$$r_{01}(w,w) = r_{10}(w,w) = r_{01}(v,v)\ (= r_{10}(v,v)\ ).$$

Next, let us consider the case where w corresponds to the output node of a logic gate with k inputs. Let v_i (i=1,2,...,k) be a vertex corresponding to an input node of the gate, and let $e_i = (v_i,w)$ be incoming edges of w. (See Fig. 3)

If the logic gate is a NAND gate, then d(w,0) is calculated by the following equation:

$$d(w,0) = \max[\ d(v_i,1) + t_1(e_i) \mid 1 \le i \le k\ ],$$

since w becomes 0 when all inputs become 1. On the other hand, since w becomes 1 when any input becomes 0, d(w,1) seems to be determined by the equation

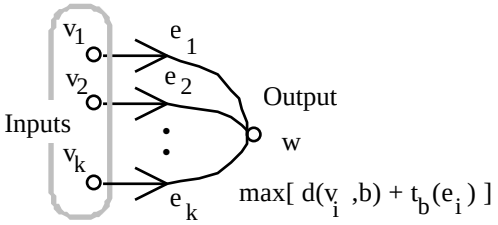

Fig. 3 A logic gate with k inputs

$$d(w,1) = \min[\, d(v_i,0) + t_0(e_i) \mid 1 \le i \le k \,].$$

However, when all inputs other than j are 1, w becomes 1 after the delay $d(v_j,0) + t_0(e_j)$. Since d(w,1) is the maximum among these delays $d(v_j,0) + t_0(e_j)$, we can calculate d(w,1) by the following equation:

$$d(w,1) = \max[\, d(v_i,0) + t_0(e_i) \mid 1 \le i \le k \,].$$

Similarly, if the logic gate is NOR gate, we can calculate d(w,0) and d(w,1) by the same maximum operations. If the logic gate is an invertor, we set k = 1 in the above equations and need not to take the maximum.

If the logic gate is a complex gate, for example, $\overline{v_1 \cdot (v_2 + v_3)}$, w becomes 0 if $v_1 = 1$ and $v_2 = 1$ or $v_3 = 1$. Therefore, d(w,0) may be determined the largest value of $d(v_i,1) + t_1(e_i)$ $(1 \le i \le 3)$. On the other hand, d(w,1) may also be determined by the largest value of $d(v_i,0) + t_0(e_i)$ $(1 \le i \le 3)$. Thus, if a complex gate consists of a series-parallel connection of MOS transistors, then we can determine d(w,b) by the same manner as NAND or NOR gate. It is not hard to see that for an AND gate or an OR gate a similar discussion holds, and d(w,b) can be computed by

$$d(w,b) = \max[\, d(v_i,b) + t_b(e_i) \mid 1 \le i \le k \,]$$

although the definition of $t_b(e)$ must be modified slightly.

For the logic gates discussed above, we only need the values of d(v,b) and $t_b(e)$, and not $d(w,\bar{b})$ and $t_b(e)$ simultaneously, where $\bar{b}$ denotes the complement of b. Therefore, we need correlations between $d(v_i,b)$ $(1 \le i \le k)$, but not correlation between d(v,b) and $d(v,\bar{b})$. However, if the logic gate is an XOR gate, then output w becomes 1 by input signal 1 or 0. Hence, d(w,1) may be equal to $d(v_i,0) + t_{01}(v_i)$ or $d(v_i,1) + t_{11}(v_i)$ $(1 \le i \le k)$, where $t_{01}(v_i)$ (or $t_{11}(v_i)$) is the switching delay between the arrival time of signal 0 (signal 1) to input v_i and the time when output w turned to 1. Therefore, d(w,1) is computed by the following equation.

$$d(w,1) = \max[\, d(v_i,0) + t_{01}(v_i),\ d(v_i,1) + t_{11}(v_i) \mid 1 \le i \le k \,].$$

Through a similar discussion, we see that d(w,0) is computed by

$$d(w,0) = \max[\, d(v_i,0) + t_{00}(v_i),\ d(v_i,1) + t_{10}(v_i) \mid 1 \le i \le k \,].$$

Thus, in the case of XOR gate, we need new edge-delays different from $t_b(e)$ and a different maximum operation. In order to simplify the descriptions, we assume in the following that a given combinatorial circuit does not contain XOR gates. Therefore, the maximum operation is conducted only among $d(\bullet,0)+t_0(\bullet)$ or among $d(\bullet,1)+t_1(\bullet)$.

III. Distribution of the Maximum

In this section, we show a method to compute mean Exp[x(b)] and variance Var[x(b)] of $x(b) = \max[\, d(v_i,b) + t_b(e_i) \mid 1 \le i \le k \,]$. This is done by procedure DISTMAX(In(w), U), where $In(w) = \{\, e_i=(v_i,w) \mid 1 \le i \le k \,\}$, and U is the set of vertices to which there is no directed path from w. The procedure first computes Exp[x(b)] and Var[x(b)] for b = 0, and then for b = 1. Moreover, it computes correlation coefficient R[x(b), d(u,b')] between x(b) and d(u,b') of vertex u U and correlation coefficient R[x(0), x(1)]. To do this, we assume that correlation coefficients $r_{bb'}(x,y)$ are known for any vertices x,y U.

According to Ref.[9], given two stochastic variables x and y with correlation coefficient R[x,y] = ρ whose distributions are $N(\mu_x, \sigma_x^2)$ and $N(\mu_y, \sigma_y^2)$, respectively, mean Exp[t] and variance Var[t] of t = max[x, y] are obtained by the following equations, unless $\sigma_1-\sigma_2 = \rho-1 = 0$;

$$\mathrm{Exp}[\,t\,] = \mu_x \cdot \Phi(\beta) + \mu_y \cdot \Phi(-\beta) + \alpha \cdot \phi(\beta),$$
$$\mathrm{Var}[\,t\,] = (\mu_1^2+\sigma_1^2)\cdot\Phi(\beta) + (\mu_2^2+\sigma_2^2)\cdot\Phi(-\beta) + (\mu_1+\mu_2)\cdot\alpha\cdot\phi(\beta) - \{\mathrm{Exp}[\,t\,]\}^2,$$

where

$$\alpha = \sqrt{\sigma_x^2+\sigma_y^2-2\sigma_x\sigma_y\rho}\,, \qquad \beta = (\mu_1-\mu_2)/\alpha,$$
$$\phi(x) = \frac{1}{\sqrt{2\pi}}\exp\left[-\frac{x^2}{2}\right],$$
$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp\left[-\frac{y^2}{2}\right]dy\,.$$

Moreover, correlation coefficient R[t,z] between t = max[x, y] and a stochastic variable z with a normal distribution can be obtained by the following equation, if correlation coefficient between x and z is $R[x,z] = \rho_x$ and that between y and z is $R[y,z] = \rho_y$.

$$R[t,z] = R[\ \max[x,y],\ z\ ] = \frac{\left[\sigma_x\cdot\rho_x\cdot\Phi(\beta) + \sigma_y\cdot\rho_y\cdot\Phi(-\beta)\right]}{\sqrt{\mathrm{Var}[\,t\,]}}.$$

We use these equations in procedure DISTMAX(In(w), U) as follows.

Let $x(b)_i^* = d(v_i,b) + t_b(e_i)$. Then, since $d(v_i,b)$ and $t_b(e_i)$ are independent, we have

$$\mathrm{Exp}[\,x(b)_i^*\,] = m_b(v_i) + \mu_b(e_i),$$
$$\mathrm{Var}[\,x(b)_i^*\,] = s_b(v_i) + \sigma^2{}_b(e_i).$$

Moreover, for $x(b)_i^*$ and stochastic variable y, the following equation holds

$$\sqrt{Var[\,x(b)_i\,]}\cdot R[\,x(b)_i^*,\,y\,] = \sqrt{s_b(v_i)}\cdot R[\,d(v_i,b),\,y\,] + \sigma_b(e_i)\cdot R[\,t_b(e_i),\,y\,] .$$

If $y = d(u,b')$ of vertex $u \in U$, then $R[\,t_b(e_i),\,y\,] = 0$. Therefore, for $u \in U$, we have

$$R[\,x(b)_i^*,\,d(u,b')\,] = \frac{\sqrt{s_b(v_i)}\cdot r_{bb'}(v_i,u)}{\sqrt{Var[\,x(b)_i\,]}} ,$$

where $r_{bb'}(v_i,u) = R[\,d(v_i,b),\,d(u,b')\,]$. If $y = t_b(e_j)$ of edge $e_j \in In(w)$, then $R[\,x(0)_i^*,\,t_1(e_j)\,] = R[\,x(1)_i^*,\,t_0(e_j)\,] = 0$, and we have

$$R[\,x(b)_i^*,\,t_b(e_j)\,] = \sigma_b(e_i)\cdot\rho_b(e_i,e_j)/\sqrt{Var[\,x(b)_i\,]} .$$

We consider correlation coefficients $R[\,x(b)_i^*,\,t_b(e_j)\,]$ only for edges $e_j \in In(w)$ such that $i<j$, since we need these coefficients only.

Now, let $x(b)_i = \max[\,d(v_j,b) + t_b(e_j) \mid 1 \le j \le i\,]$. Then, since $x(b)_1 = x(b)_1^*$, for $i \ge 2$, we can compute $Exp[\,x(b)_i\,]$, $Var[\,x(b)_i\,]$, $R[\,x(b)_i,\,d(u,b')\,]$, and $R[\,x(b)_i,\,t_b(e_j)\,]$ by using

$$x(b)_i = \max[\,x(b)_{i-1},\,x(b)_i^*\,]$$

and the equations in Ref.[9]. Namely, let

$$\alpha = \sqrt{Var[\,x(b)_{i-1}\,] + Var[\,x(b)_i\,] - 2\gamma} ,$$

$$\gamma = \sqrt{Var[\,x(b)_{i-1}\,]}\cdot\sqrt{Var[\,x(b)_i\,]}\cdot R[\,x(b)_{i-1},\,x(b)_i\,]$$

and

$$\beta = (\,Exp[\,x(b)_{i-1}\,] - Exp[\,x(b)_i^*\,]\,)/\alpha ,$$

where

$$R[\,x(b)_{i-1},x(b)_i^*\,] = \frac{\sqrt{s_b(v_i)}\cdot R[\,x(b)_{i-1},\,d(v_i,b)\,] + \sigma_b(e_i)\cdot R[\,x(b)_{i-1},\,t_b(e_i)\,]}{\sqrt{Var[\,x(b)_i\,]}} .$$

Then, we can compute $Exp[\,x(b)_i\,]$, $Var[\,x(b)_i\,]$, and $R[\,x(b)_i,\,d(u,b')\,]$ for $u \in U$ as follows.

$$Exp[\,x(b)_i\,] = Exp[\,x(b)_{i-1}\,]\cdot\Phi(\beta) + Exp[\,x(b)_i^*\,]\cdot\Phi(-\beta) + \alpha\cdot\phi(\beta) ,$$

$$Var[\,x(b)_i\,] = \{(Exp[\,x(b)_{i-1}\,])^2 + Var[\,x(b)_{i-1}\,]\}\cdot\Phi(\beta) + \{(Exp[\,x(b)_i^*\,])^2 + Var[\,x(b)_i^*\,]\}\cdot\Phi(-\beta) + \{Exp[\,x(b)_{i-1}\,] + Exp[\,x(b)_i'\,]\}\cdot\alpha\cdot\phi(\beta) - \{Exp[\,x(b)_i\,]\}^2,$$

$$R[\,x(b)_i,\,d(u,b')\,] = \frac{X_u + Y_u}{\sqrt{Var[\,x(b)_i\,]}} ,$$

$$X_u = \sqrt{Var[\,x(b)_{i-1}\,]}\cdot R[\,x(b)_{i-1},\,d(u,b')\,]\cdot\Phi(\beta) ,$$

$$Y_u = \sqrt{Var[\,x(b)_i\,]}\cdot R[\,x(b)_i,\,d(u,b')\,]\cdot\Phi(-\beta) .$$

Correlation coefficient $R[\,x(b)_i,\,t_b(e_j)\,]$ $(j > i)$ can be computed as follows.

$$R[\,x(b)_i,\,t_b(e_j)\,] = \frac{X_e + Y_e}{\sqrt{Var[\,x(b)_i\,]}} ,$$

$$X_e = \sqrt{Var[\,x(b)_{i-1}\,]}\cdot R[\,x(b)_{i-1},\,t_b(e_j)\,]\cdot\Phi(\beta) ,$$

$$Y_e = \sqrt{Var[\,x(b)_i\,]}\cdot R[\,x(b)_i,\,t_b(e_j)\,]\cdot\Phi(-\beta) .$$

Repeating the computations stated above for i=2,3,...,k, we have the distribution of x(b), since $x(b) = x(b)_k$. Hence, by conducting this repetition for b = 0, we have $Exp[\,x(0)\,]$, $Var[\,x(0)\,]$, and $R[\,x(0),\,d(u,b)\,]$ for $u \in U$. Then, conducting the repetition for b = 1, we obtain $Exp[\,x(1)\,]$, $Var[\,x(1)\,]$, and $R[\,x(1),\,d(u,b)\,]$. In this case, however, we need not to compute $R[\,x(1)_i,\,d(v_j,0)\,]$ for v_j such that $e_j=(v_j,w) \in In(w)$. Because, we do not use these values. But, we compute $R[\,x(1)_i,\,x(0)\,]$ as follows. Since $R[\,t_1(e_i),\,x(0)\,] = 0$, we have

$$R[\,x(1)_i^*,\,x(0)\,] = \frac{\sqrt{s_1(v_i)}\cdot R[\,d(v_i,1),\,x(0)\,]}{\sqrt{Var[\,x(1)_i\,]}} ,$$

and

$$R[\,x(1)_i,\,x(0)\,] = \frac{X_{01} + Y_{01}}{\sqrt{Var[\,x(1)_i\,]}} ,$$

$$X_{01} = \sqrt{Var[\,x(1)_{i-1}\,]}\cdot R[\,x(1)_{i-1},\,x(0)\,]\cdot\Phi(\beta) ,$$

$$Y_{01} = \sqrt{Var[\,x(1)_i\,]}\cdot R[\,x(1)_i,\,x(0)\,]\cdot\Phi(-\beta) .$$

Hence, we have $R[\,x(1),\,x(0)\,] = R[\,x(1),\,x(0)\,] = R[\,x(1)_k,\,x(0)\,]$.

IV. Algorithm

In this section, we describe the proposed algorithm by using procedure DISTMAX(In(w), U) shown in the previous section.

Our problem is to find mean $m_b(v)$ and variance $s_b(v)$ of d(v,b) for each sink $v \in T$ and correlation coefficient $r_{bb'}(v,w)$ for each pair of sinks $v, w \in T$. Once these values are obtained, we can compute the probability for the maximum delay $M = \max[\,d(v,b) \mid v \in T,\, b \in \{0,1\}\,]$ to be not greater than D, from the following equation

$$Pro[\,M \le D\,] = \int_{-\infty}^{D}\int_{-\infty}^{D}\cdots\int_{-\infty}^{D} f(x_1,x_2,\ldots,x_n)\,dx_1\,dx_2\ldots dx_n$$

where $n = 2|T|$, each x_i corresponds to d(v,b), and $f(x_1, x_2,\ldots,x_n)$ is the probability density function of normal distribution with $2|T|$ variables. If we need its probability density function g(D) of distribution $G(D) = Pro[\,M \le D\,]$, it can be obtained by differentiating G(D). Although mean $Exp[\,g(D)\,]$ and variance $Var[\,g(D)\,]$ of the maximum delay M are computed numerically, they can be computed by a method similar to procedure DISTMAX by assuming g(D) as

a normal distribution. The outline of the proposed algorithm is as follows.

We first reduce a given circuit graph G=(V,E) by repeating the reduction of two series edges e'=(u,v) and e''=(v,w) into one edge e*=(u,w). Since the delays of two series edges are independent, the mean and variance of $t_b(e^*)$ are given by

$$\mu_b(e^*) = \mu_b(e') + \mu_b(e''),$$
$$\sigma_b^2(e^*) = \sigma_b^2(e') + \sigma_b^2(e''),$$

respectively, and correlation coefficient $\rho_b(e^*,e)$ of $t_b(e^*)$ with $t_b(e)$ of edge e coming into w is given by

$$\rho_b(e^*,e) = \frac{\sigma_b(e')\bullet\rho_b(e',e)+\sigma_b(e'')\bullet\rho_b(e'',e)}{\sqrt{\sigma_b^2(e')+\sigma_b^2(e'')}},$$

in these equations, if $\sigma_b(\bullet) = 0$, then let $\rho_b(\bullet,\bullet) = 0$.

Then, we determine $m_b(v)$ and $s_b(v)$ of each vertex v of the reduced graph G*=(V*,E*) in the topological order with the use of procedure DISTMAX as follows. Let *Front* be a set of vertices satisfying the following conditions.

(A) For any vertex v *Front*, $m_0(v)$ and $s_0(v)$ of d(v,0) and $m_1(v)$ and $s_1(v)$ of d(v,1) are known.

(B) For any pair of vertices u, v *Front*, $r_{00}(u,v)$, $r_{01}(u,v)$, $r_{10}(u,v)$, and $r_{11}(u,v)$ are known.

Initially, set *Front* as the set of vertices corresponding to primary inputs, and we see that conditions (A) and (B) can be satisfied for this *Front*. Then, repeat the following procedure, until *Front* becomes T.

1°: select a vertex w *Front* such that all incoming edges of w come from vertices in *Front*;

2°: let *Del*(w) be the set of vertices v *Front* such that all outgoing edges (v,u) from v come into vertices u *Front* {w};

3°: set *Front* := *Front* - *Del*(w);

4°: conduct procedure DISTMAX(In(w), *Front*), and determine $m_b(w)$, $s_b(w)$, and $r_{bb'}(w,u)$ for u *Front*, according to the type of the gate containing w as its output terminal. Namely, for example, if the gate is AND gate, then set $m_b(w)$=Exp[x(b)] and so on. Moreover, since we have R[x(0), x(1)], we can determine $r_{bb'}(w,w)$ as follows:

$$r_{00}(w,w) = r_{11}(w,w) = 1,$$
$$r_{01}(w,w) = r_{10}(w,w) = R[\, x(1), x(0) \,].$$

5°: add w to *Front*;

If *Front* becomes T, we compute Exp[M_b] = Exp[max[d(v,b) | v T]], Var[M_b], and R[M_0, M_1], with the use of procedure DISTMAX({ (u,t) | u T }, T), under the assumption that the delay of all edges (u,t) are constant and equal to 0. Then, we compute Exp[M] = Exp[M_0, M_1] and Var[M] by the equations in [9].

The time complexity of the proposed algorithm can be analyzed as follows.

Let k_w be the number of in-coming edges of a vertex w V-{v_0}, and h_{max} be the maximum number of vertices contained in *Front* simultaneously. Then, the time required for adding a vertex w to *Front* can be denoted by $O(k_w\bullet(h_{max}+k_w))$, since for each edge $e_i = (v_i,w)$, correlation coefficients R[$x(b)_i$, d(u,0)] and R[$x(b)_i$, d(u,1)] are computed for each u *Front*, and R[$x(b)_i$, $t_b(e_j)$] for each edge $e_j=(v_j,w)$ In(w) such that j>i. Since the maximum value of k_w corresponds to the maximum number of inputs of a logic gate, we may assume $h_{max} \geq k_w$ so that the time needed in addition of a vertex w to *Front* is $O(k_w\bullet h_{max})$. Therefore, the total time complexity is $O(m\bullet h_{max})$, since no vertex is added to *Front* more than once and $\sum_w{}_V k_w \leq m$, where m = |E| is the number of edges of a given circuit graph G=(V,E).

In the worst-case, the maximum size h_{max} of *Front* is O(n), where n = |V| is the number of vertices. However, in real circuits, h_{max} is smaller than n, so that the time complexity of the algorithm is less than $O(n\bullet m)$.

V. Experimental Results

In order to see the performance of the proposed algorithm, we first applied our algorithm to the graph in Fig. 4, which corresponds to the circuit in Fig. 1. The mean μ and standard deviation σ of delays of edges (a,x) and (c,y) are 20 and 1.4, respectively, and those of edges (v,x), (v,y), (x,z), and (y,z) are 10 and 0.7, respectively. We changed the delay of edge (u,v), and computed the maximum delay to vertex z, as shown in Table 1. In the experiments, we assumed that all edge-delays are independent.

In the table, "Monte Carlo," "Ours," and "No Correlation" show the results obtained by Monte Carlo simulation (20,000 iterations), by our algorithm, and by ignoring correlation of delays d(x,b) and d(y,b), respectively, and "error" is the relative error to the result of Monte Carlo simulation. We showed in the table correlation coefficient r(x,y) and cpu-time spent in ULTRA-SPARC IIi 400MHz workstation, too. Fig. 5 shows the distributions of the maximum delay d(z) in the case of the top row in the table. Although the difference between "Ours" and "No Correlation" may not be large, the difference increases if σ becomes large.

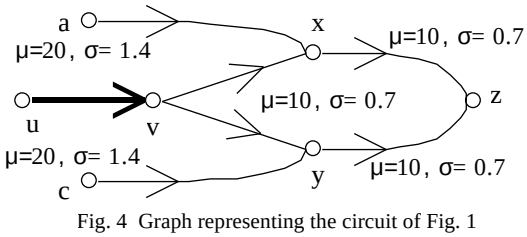

Fig. 4 Graph representing the circuit of Fig. 1

TABLE 1 DELAYS OF EDGE (u,v) AND THE MAXIMUM DELAY d(z)

	delay of edge (u,v)	maximum delay d(z)			r(x,y)
		Monte Carlo	Ours (error)	No Correlation (error)	
μ	80	100.0	100.6 (0.54%)	103.2 (3.19%)	0.980
σ	5.6	5.67	5.66 (-0.22%)	4.70 (-17.2%)	
cpu [sec]		4.48	0.67	0.27	
μ	40	60.27	60.56 (0.48%)	61.68 (2.34%)	0.941
σ	2.8	2.90	2.92 (0.54%)	4.70 (-15.5%)	
cpu [sec]		4.10	0.52	0.21	
μ	20	40.36	40.56 (0.48%)	40.97 (1.49%)	0.800
σ	1.4	1.63	1.62 (-0.27%)	1.42 (-12.9%)	
cpu [sec]		3.91	0.29	0.18	
μ	10	31.21	31.34 (0.42%)	31.37 (0.52%)	0.122
σ	0.7	1.07	1.03 (-4.26%)	1.01 (-6.05%)	
cpu [sec]		3.92	0.19	0.16	

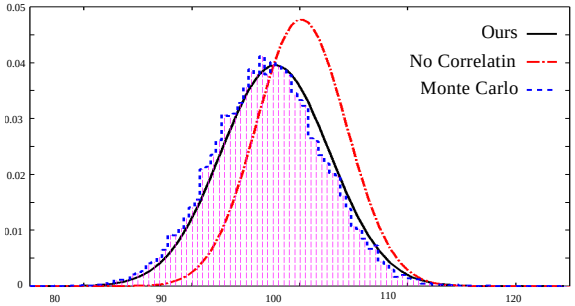

Fig. 5 The distributions of maximum delays

TABLE 2 RESULTS OF ISCAS85 BENCHMARK DATA

		Ours	No Correlation (error)
c432 n = 541 m = 722	μ	6.64 nsec	6.70 nsec (0.92%)
	σ	0.29 nsec	0.28 nsec (-6.23%)
	cpu time	107.0 sec	3.7 sec
c499 n = 685 m = 921	μ	9.98 nsec	10.10 nsec (1.18%)
	σ	0.28 nsec	0.205 nsec (-28.4%)
	cpu time	208.4 sec	6.5 sec
c880 n = 1,200 m = 1,570	μ	14.64 nsec	14.73 nsec (0.67%)
	σ	0.58 nsec	0.55 nsec (-5.07%)
	cpu time	371.7 sec	6.8 sec

Table 2 shows the results of a few circuits in ISCAS 85 benchmark, where n and m are the numbers of vertices and edges of the graph representing the circuits, respectively, and error is the percentage of the difference between "Ours" and "No Correlation." In the experiments, the circuits are laid out by a macrocell layout system[8], and the mean values of delays are extracted from the layout. The standard deviation σ is set as σ=0.15μ, and the edge delays are assumed to be independent again. The cpu-time is mainly depend on the number of numerical computations for normal distribution, and hence it can be reduced by using table look-up technique. Thus, further investigation about correlation and reduction of CPU time are future work.

VI. CONCLUSION

In this paper, we proposed a new algorithm for the statistical static timing analysis which can treat correlations of delays of re-convergent paths and correlations of delays of adjacent transistors. Since the algorithm takes the correlations into account, the worst-case time complexity is O(n•m), where n and m are the numbers of vertices and edges of a given acyclic graph representing a CMOS combinatorial circuit, respectively.

In order to see effects of correlation, we applied our algorithm to some benchmark circuits, and found that the difference between the results obtained by our algorithm and those obtained by ignoring correlations is not large enough, comparing to the increase of the computational effort. However, since the difference increaaes, if the distribution becomes large, further research is necessary.

There still remain a few problems on the statistical static timing analysis, among which one of the most important issues is the deletion of false paths[12]. We are tackling this problem by using the proposed technique.

ACKNOWLEDGEMENT

The authors express their gratitude to Prof. T. Nakano, Dept. of Physics, Chuo University, for his valuable discussions about calculations of normal distributions, and Mr. Shuji Nishimoto, Dept. of EECE, Chuo University, for his programming of the proposed algorithm and collection of experimental results.

REFERENCES

[1] H.-F. Jyu, S. Malik, S. Devadas, K.W. Keutzer, "Statistical timing analysis of combinatorial logic circuits," IEEE Trans. VLSI Systems, vol.1, no.2, pp.126-137, 1993

[2] L. Rung-Bin and W. Meng-Chiou, "A new statistical approach to timing analysis of VLSI circuits," Proc. 11th Int. Conf. on VLSI Design, pp.507-513, 1997.

[3] M.Berkelaar, "Statistical delay calculation, a linear time method," Proc. Int. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU97), pp.15-24, 1997.

[4] H. Matsunaga, D. Mizoguchi, H. Yasuura, "Estimation of combinatorial circuit delays using the distribution of CMOS gate delays," Proc. DA Symp. '99, pp.77-82, 1999. (in Japanese)

[5] S. Tsukiyama, M. Tanaka, M. Fukui, "An estimation algorithm of the critical path delay for a combinatorial circuits," Proc. the 13th Workshop on Circuits and Systems in Karuizawa, pp.131-136, 2000. (in Japanese)

[6] M.Hashimoto and H.Onodera, "A performance optimization method by gate resizing based on statistical static timing analysis," Proc. Workshop on Synthesis And System Integration of MIxed Technology (SASIMI 2000), pp.77-82, 1999.

[7] E.T.A.F.Jacobs and M.R.C.M.Berkelaar, "Gate sizing using a statistical delay model," Proc. Design Automation and Test in Europe (DATE2000), pp.283-290, 2000.

[8] T. Tamai, S. Nishimoto, S. Tsukiyama, M. Tanaka, M. Fukui, "A layout optimization system of macrocells," Tech. Report of IEICE, VLD 99-128, pp.85-91, 2000. (in Japanese)

[9] C.E. Clark, "The greatest of a finite set of random variables" Operations Research, vol.9, pp.145-152, 1961.

[10] M. Kondo, H. Onodera, K. Tamaru, "MOSFET statistical modeling method using an intermediate model," Trans. of IEICE A, vol.J81-A, no.11, pp.1555-1563, 1998. (in Japanese)

[11] R. Kay and L. Pilleggi, "PRIMO: Probability interpretation of moment for delay calculation", Proc. Design Automation Conf., pp.463--468, 1998.

[12] H.C. Chen and D.H. Du, "Path sensitization in critical path problem," IEEE Trans. Computer-Aided Design of ICs and Systems, vol.12, no.2, pp.196-207, 1993.