Algorithm for Fast Statistical Timing Analysis

Jakob Salzmann[1], Frank Sill[2], Dirk Timmermann[1]

[1]University of Rostock
Rostock, Germany
Department of CS and EE
{jakob.salzmann, dirk.timmermann}@uni-rostock.de

[2]Federal University of Minas Gerais (UFMG)
Belo Horizonte, Brazil
Department of Electrical Engineering
frank@ufmg.br

***Abstract*—Problems of parameter variations are a main topic in current research and will gain importance in future technology generations due to the continuing scaling. Therefore, it requires appropriate timing analysis which is traditionally done with corner-case simulations. These are quite conservative and pessimistic approaches. In contrast, new statistical static timing analysis (SSTA) algorithms offer a more accurate prediction of the timing behavior of circuit designs. Further, correlations between various parameters and devices can be observed. Unfortunately, the SSTA algorithms mostly require high computational effort and accurate library characterization. This paper proposes an approach for a fast statistical static timing analysis (F-SSTA) with moderate requirements on computation time and library characterization. The approach considers the analysis of gates with multiple inputs. The simulation results show an average error of 5 % compared to Monte-Carlo simulations but a significant speed improvement of around 20 times compared to a highly accurate SSTA algorithm.**

I. Introduction

Aggressive downscaling of CMOS devices in each technology generation results in rising integration density and performance. At the same time, the influence of parameter variations also increases drastically [1]. This is due to different effects as fluctuations of process parameters, temperature, or supply voltage. As a consequence, changes occur in transistor characteristics, which might cause longer delay and higher power dissipation. This again results in a high uncertainty about the design and manufacturing conditions which is the reason for unnecessary over-design and underperformance of circuits [2,3].

Hence, statistical static timing analysis (SSTA) has been recommended by a great amount of researchers over the past years [3]-[7]. SSTA offers an accurate characterization of timing behavior. Further, correlations between varying parameters and gates can be observed. In [3], Orshansky presents a general framework for SSTA. Agarwal et al. propose in [4] approaches to handle SSTA on multiple input gates.

Three of the main problems of current SSTA algorithms are high characterization efforts for the gates, long run-times, and the data-dependent timing of gates with multiple inputs. The purpose of this paper is the presentation of a fast SSTA (F-SSTA) algorithm, with only moderate deviations compared to results from Monte-Carlo simulations. Section 2 introduces the differences between common deterministic and new statistical timing analyses. Section 3 presents an approach for a simplified gate delay characterization under several parameter variations. Sections 4 and 5 propose a method to handle correlations between varying gates, and a method for timing analysis of gates with multiple inputs, respectively. Finally, section 6 contains a comparison of the proposed F-SSTA and a highly accurate SSTA, while section 7 draws the conclusion.

II. Timing Analysis

A. Corner-case Static Timing Analysis

The concern of worst-case static timing analysis (STA) is the evaluation of the guaranteed circuit performance. This knowledge is necessary to integrate circuits into complex design environments. STA can be performed at different design levels but the timing analysis at gate level offers the best trade-off between evaluation time and accuracy.

Corner-case STA is a common approach to handle parameter variations. This approach applies gate libraries with corner-case models. This means, each gate is characterized for most important parameter sets to extract its behavior for typical, worst and best conditions. At timing analysis, signal arrival times of the gate outputs are estimated by adding the gate delay to the signal arrival time at the inputs. Thereby, the worst-case design delay is estimated for each gate set to its maximum delay value.

In current and future technologies with heavy parameter variations corner-case STA results in very pessimistic prediction of performance [2,3,5]. One reason is that STA assumes all parameters of all gates in worst-case at the same time. However, the amount of parameters which influence the delay is growing. Hence, the worst-case probability decreases. Thus, the designs are produced for a case which has a very low probability. Further, STA leads to an underestimation of performance. This is based on the assumption of perfect correlation of delay response sensitivity of each gate to variation in each parameter with all other delay elements [3]. Additionally, STA completely ignores intra-die variations [4].

B. Statistical Static Timing Analysis

Statistical static timing analysis (SSTA) offers a probability based prediction of timing behavior and taking intra-die variations into consideration. Gate delay t_d of SSTA is described as a probabilistic function which is mostly formulated as Gaussian distribution. Hence, signal arrival times are also modeled as probabilistic functions. The delay variability can be described with probability density functions (PDF) or cumulative probability distribution functions (CDF). The PDF is the probability

that a signal arrival time or a gate delay has the value *t*. In contrast, a CDF describes the probability that the signal arrival time or the gate delay is lower than a given value *t*. Thus, the CDF is equal to the probability density. It is:

$$PDF(t) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \mathrm{e}^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} \tag{1}$$

$$CDF(t) = \int_0^t \frac{1}{\sigma\sqrt{2\pi}} \cdot \mathrm{e}^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx \tag{2}$$

Here, μ is the mean value and σ is the standard deviation. For a simple Inverter the arrival time t_{out} of the output signal results from:

$$\mu_{tout} = \mu_{tin} + \mu_{Inverter} \tag{3}$$

$$\sigma_{tout} = \sqrt{\sigma_{tin}^2 + \sigma_{Inverter}^2} \tag{4}$$

Here, μ_{tout} and μ_{tin} are the mean values of the arrival times of the output and the input signal. Further, σ_{tout} and σ_{tin} are the standard deviations of both arrival times. Additional, $\mu_{Inverter}$ and $\sigma_{Inverter}$ are the mean value and the standard deviation of the Inverter delay, respectively.

C. *Monte-Carlo simulations*

If a random parameter set is applied for the system simulation this is referred as Monte-Carlo simulation [3]. That means for timing analysis every gate is defined with different parameters which have an influence on gate delay. A sufficient amount of Monte-Carlo simulations (here: 1000) of the same system allows a very accurate determination of the probability behavior of the timing. Unfortunately, the expenditure of time for Monte-Carlo simulations is very high.

III. MODELING OF GATE VARIATIONS

The delay t_d of a CMOS device can be modeled with the alpha-power law model as [6]:

$$t_d = \frac{k' \cdot V_{dd} C_L}{\left(W / L_{eff}\right) \cdot \left(V_{dd} - V_{th}\right)^{\alpha}} \tag{5}$$

where k' is a technology constant, V_{dd} is the supply voltage, W and L_{eff} are gate width and effective length, respectively, C_L is the load, and α models the short channel effects. The threshold voltage V_{th} can be modeled as:

$$V_{th} = \frac{q}{nk_bT}\left(V_{th0} + \gamma'\sqrt{N_{DEP}}T_{ox}V_{bs} + \eta'\frac{T_{ox}}{L_{eff}^2\sqrt{N_{DEP}}}V_{ds}\right) \tag{6}$$

Here, T is the operating temperature, n is the sub-threshold swing coefficient, V_{th0} is the zero-bias threshold voltage, V_{bs} is the bulk-source voltage, V_{ds} is the drain-source voltage, η' models the drain induced barrier lowering effect, and γ' regards the body-bias effect. The terms q and k_b correspond to physical constants (electron charge, and Boltzmann's constant, respectively). N_{DEP} labels the channel doping concentration, and T_{ox} the thickness of the oxide layer. Except for the physical constants all other parameters can vary. From these, the parameters N_{DEP}, T_{ox}, L_{eff}, and W have the highest influence on delay [7].

To model parameter variations common approaches vary technology or transistor parameters like gate length L_{eff} or gate width W which strongly impact gate delay [7][8]. Then, gate delay is described as a function of the varying parameters. This allows an accurate mathematical formulation of the problem. But, computational effort increases drastically with each additional parameter. Thus, for a fast timing analysis an easy to handle gate delay model is required which considers all possibly varying parameters with low computational effort.

All varying parameters which correspond to the gate delay can be described as Gaussian distributions. This bases on the fact that the variations are expected to be truly random in nature [9]. Hence, the multiplications in (5) turn into convolutions of Gaussian distributions. As the convolution of Gaussian distributions results in new Gaussian distributions the gate delay t_d can be approximated as Gaussian distribution with:

$$P(t_d) = \frac{1}{\sigma_G\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{t_d-\mu_G}{\sigma_G}\right)^2} \tag{7}$$

$P(t_d)$ is the probability that the gate delay has the value t_d. The parameters μ_G and σ_G result from Monte-Carlo transistor level simulations of the gates. Thereby, the values of all varying parameters are formulated as Gaussian distributions. Thus, this approach applies the high accuracy of transistor simulation models. Further, the influence of all varying parameters can be observed with constant effort for the analysis. Figure 1 depicts the results of a Monte-Carlo analysis and the proposed approach for an inverter and a NAND2 gate. The parameters L_{eff}, W, N_{DEP}, and T_{ox} of each transistor are assumed to show 10 % variation. The results indicate that the differences between Monte-Carlo transistor level simulations and the Gaussian gate delay model are very small.

This gate level approach has three drawbacks. First, it is not possible to change the area in which the parameter varies. However, without the temperature all parameters variations are technology dependent and consequently fixed. Thus, a temperature factor should be explored in future works. The second drawback regards the load and the input slopes with which the gate is simulated. Thus, as in standard gate libraries the gates have to be characterized for different loads and input slopes. The last drawback is the observation of inter-gate and inter-parameter correlations, respectively. Thus, the next section proposes a solution for handling correlations between gates.

IV. TIMING ANALYSIS OF MULTI-INPUT GATES

Next, the proposed approach is extended to multi-input switching (MIS). A very common approach for evaluating output signal arrival time at multi-input gates is the creation of tables which include the results for different combinations of input signal arrival times [9]. In [4] gates with multiple inputs are

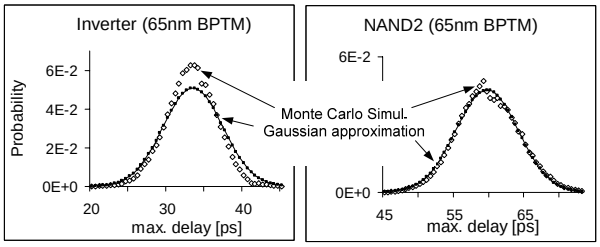
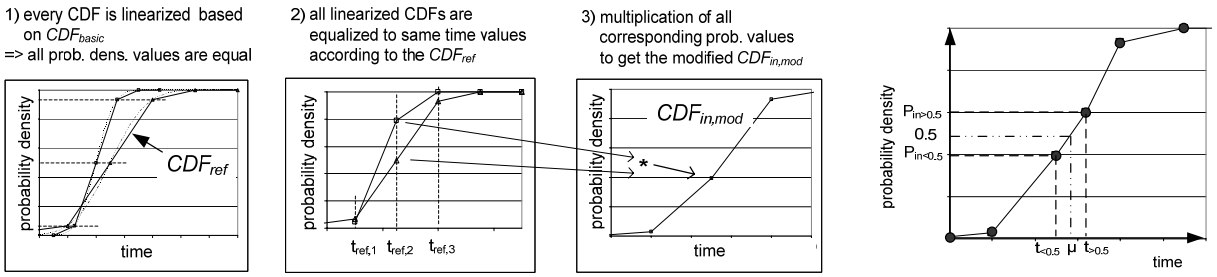


Figure 1. Gaussian description of gate delay compared to Monte-Carlo simulations



(a) (b)

Figure 2. Extraction of the modified input function *CDFin,mod* (a) and estimation of the mean value μ of *CDFin,mod* (b)

divided into single input gates. Both approaches offer good results. However, both techniques considerably increase complexity or require extensive library characterization. Thus, the following approach combines good accuracy and low efforts in characterization and complexity.

The arrival time of the output signal of a multi-input gate's results from the convolution of the input signal PDFs or the multiplication of the input signal CDFs. Unfortunately, both operations do not result in a Gaussian distribution. Hence, the output signal can only be approximated. The new approach proposes a merging of the CDFs of the arrival times of the input signals. This is done by a piecewise linearization and multiplication of all inputs CDF. Thus, based on the values of μ and σ the input signal CDFs are characterized with some pairs of time values t_x and its corresponding probabilities $P(t_x)$. These points are connected by linear functions $y = mx + n$.

It can be observed that the shape of a CDF depends on σ only. Thus, the rising factor m of a linear function between two points can be modeled as a constant ψ divided by σ:

$$m = \frac{P(t_2) - P(t_1)}{t_2 - t_1} = \frac{\psi}{\sigma} \tag{8}$$

As the n factor of the linear functions is directly connected to the mean value μ a basic CDF_{basic} has to be characterized only. From this, the piecewise approximation of all input's CDF can be extracted with the standard deviation σ of the CDFs.

The function points of the output signal's CDF result from the multiplication of these approximations. It has to be observed that all CDFs are characterized at same probability values (like P = 0.1, P = 0.5, etc.) but different time values. However, only points with equal time values t_x can be multiplied. This demands an equalization of the time values of the piecewise modeled CDFs by linear approximation. The procedure is as follows:

1) Based on the basic function CDF_{basic} every input function CDF_{in} is separated in linear parts. The points results from:

$$P_{CDFin,x}(t_{CDFin,x}) = P_{CDFbasic,x}(t_{CDFbasic,x}) \tag{9}$$

$$t_{CDFin,x} = \mu_{CDFin} + \sigma_{CDFin} \cdot t_{CDFbasic,x} \tag{10}$$

Here, $t_{CDFin,x}$ are the time values of *CDFin* which has the same probability density values as the time values $t_{CDFbasic,x}$ of the basic function.

2) All corresponding points of the input functions are equalized to same t_x values. Therefore, the CDF with the highest t value at a probability density of $P(t) = 1$ is the reference function CDF_{ref}. The linear approximation of the new points of the other CDFs occurs with:

$$P_{new}(t_{ref}) = \frac{t_{ref} - t_{lo}}{t_{hi} - t_{lo}} \cdot \left[P(t_{hi}) - P(t_{lo})\right] + P(t_{lo}) \tag{11}$$

Here, t_{ref} is the time value of CDF_{ref}. $P_{new}(t_{ref})$ is the approximated probability density value of a CDF_{in} at t_{ref}. t_{lo} is the time value of the CDF_{in} which is the closest lower value than t_{ref}. In contrast, t_{hi} is the closest larger time value than t_{ref}. $P(t_{lo})$ and $P(t_{hi})$ are the probability density values which correspond to the time values t_{lo} and t_{hi}.

3) All corresponding probability density values of the input CDFs are multiplied to get a new modified input $CDF_{in,mod}$. This means for all time points which are modeled by the reference CDF_{ref}:

$$P_{CDFin,mod}(t_{ref}) = \prod_{all\_CDF_{in}} P_{CDF,i}(t_{ref}) \tag{12}$$

Figure 1 depicts an example for the extraction of a merged $CDF_{in,mod}$ of two CDFs. The result is a piecewise linear $CDF_{in,mod}$ which results from the multiplication of arrival times of all input signals. As next, μ_{in} and σ_{in} has to be extracted. In a CDF, the mean value μ is equal to the time value which has a probability density of 0.5. That means:

$$P(t_\mu) = 0.5 \quad \text{with } t_\mu = \mu \tag{13}$$

t_μ is determined by the approximation of the piecewise linear description of the output function. Thus, the algorithm continues as follows:

4) Linear approximation of the expected value μ_{out} of the output signal's arrival time:

$$\mu_{in,mod} = \frac{0.5 - P_{in<0.5}(t_{<0.5})}{P_{in>0.5}(t_{>0.5}) - P_{in<0.5}(t_{<0.5})}\left(t_{>0.5} - t_{<0.5}\right) + t_{<0.5} \tag{14}$$

Here, μ_{in} is the mean value of the modified input signal's $CDF_{in,mod}$. $P_{in<0.5}$ and $P_{in>0.5}$ are the probability density values of $CDF_{in,mod}$ which are close to P = 0.5. $t_{<0.5}$ and $t_{>0.5}$ are the corresponding time values. Next, the estimation of the standard deviation σ_{in} of $CDF_{in,mod}$ is extracted by the estimation of relatively standard deviation σ_{rel} for each pair of the piecewise linear $CDF_{in,mod}$. Thereby, each σ_{rel} value is determined relatively to the basic CDF_{basic}. The next steps are:

5) For every probability density value which is determined in CDF_{basic} the corresponding time value $t_{in,x}$ of $CDF_{in,mod}$ is

approximated. This is done like in step 4. Then, a relative time value t_{rel} is estimated for every $t_{in,mod,x}$ value with:

$$t_{rel,x} = t_{in,x} - \mu_{in} \quad (15)$$

6) Based on these values, for each pair a standard deviation value $\sigma_{rel,x}$ which is relatively to the standard deviation σ_{basic} of CDF_{basic} can be estimated. This means:

$$\sigma_{rel,x} = \frac{t_{rel,x}}{t_{basic,x}} \sigma_{basic} \quad (16)$$

Here, $t_{basic,x}$ is the time value which has the same probability density as to $t_{rel,x}$.

7) The standard deviation σ_{in} of the modified arrival time of the input signal is the medium value of all $\sigma_{rel,x}$. Hence:

$$\sigma_{in} = \frac{\sum_{n_\sigma} \sigma_{rel,x}}{n_\sigma} \quad (17)$$

Here, n_σ is the number of $\sigma_{rel,x}$ values different from 0. After the extraction of μ_{in} and σ_{in} the arrival time of the output signal of the multi-input gate can be estimated as for an Inverter (see equ. 3 and 4).

V. Simulation results

The new F-SSTA algorithm was tested with different configurations of a ten gates deep tree circuit. The applied gates base on predictive technology models (BPTM) [10]. Thereby, transistor parameters N_{DEP}, T_{ox}, L_{eff}, and W were modeled as Gaussian distribution (σ = 5% of μ). All gates were characterized using the proposed method (see section 3). The new algorithm is compared with the SSTA algorithm from Agarwal et al. [4] (named AG-SSTA in the following). Thus, the worst-case delay of the chain was estimated with Monte Carlo simulations, with the proposed F-SSTA algorithm, with the AG-SSTA algorithm, and with a common STA algorithm (*System: 1.8 GHz AMD, Suse Linux 10.0, Java 1.4*). At every simulation the $delay_{99.9\,\%}$ value was estimated. This value indicates when the design's maximum delay is with a probability of 99.9 % lower than $delay_{99.9\%}$. Thereby, the $delay_{99.9\%}$ value of the Monte Carlo simulations is the reference. Hence, the differences between the $delay_{99.9\%}$ values of the Monte Carlo simulation and the (S)STA algorithms indicate the error of the (S)STA algorithms. A positive error means an overestimation while a negative error shows an underestimation of the delay.

The analysis of the tree circuit reveals the accuracy of the proposed algorithm. Thereby, the maximum error of the F-SSTA algorithm is 8 % while the average error is 5 %. In contrast, the AG-SSTA algorithm has an average error of -3 %. This underestimation should not happen as this failure prediction can decrease the reliability of the designs. Finally, the STA algorithm has a maximum error of 20 % and an average error of 16 %. The comparison of all runtimes indicates STA as the fastest algorithm with an average runtime of 60 ms. However, the average runtime of the proposed algorithm is with 73 ms only slightly longer. In contrast the average runtime of the AG-SSTA (1942 ms) is more than 20 times longer. Hence, the proposed F-SSTA algorithm offers are very good tradeoff between evaluation time and accuracy. It has to be noted that the error is always positive. Thus, the calculated delay of the F-SSTA algorithm is overestimated which is a prerequisite for a reliable design.

TABLE I. Simulation results of a tree circuit whereas the $DELAY_{99.9\%}$ value is the delay with a probability of 99.9% and the $ERROR_{99.9\%}$ value is the difference of the $DELAY_{99.9\%}$ value compared to the value of the Monte-Carlo simulations. The different cases represent different combinations of data arrival times for the inputs of the tree structure.

		Case 1	Case 2	Case 3
Monte-Carlo	μ	1052 ps	975 ps	1001 ps
	σ	32 ps	35 ps	38 ps
	$delay_{99.9\%}$	1148 ps	1082 ps	1115 ps
	Time	20 min	21 min	19 min
F-SSTA	μ	1082 ps	1039 ps	1039 ps
	σ	33 ps	42 ps	42 ps
	$delay_{99.9\%}$	1182 ps	1166 ps	1166 ps
	$error_{99.9\%}$	3.0 %	7.8 %	4.6 %
	Time	70 ms	80 ms	70 ms
AG-SSTA [4]	μ	1044 ps	969 ps	989 ps
	σ	27 ps	27 ps	27 ps
	$delay_{99.9\%}$	1125 ps	1050 ps	1070 ps
	$error_{99.9\%}$	-2.0 %	-3.0 %	-4.2 %
	Time	1423 ms	1421 ms	1482 ms
STA	*T*	1295 ps	1295 ps	1295 ps
	$error_{99.9\%}$	13 %	20 %	16 %
	Time	60 ms	60 ms	60 ms

VI. Conclusions

This paper proposes the F-SSTA algorithm for fast statistical static timing analysis to handle parameter variations in upcoming technologies. It applies a Gaussian description of all gate delays which results from Monte Carlo simulations on transistor level. Thus, all varying parameters can be observed. Further, the algorithm offers the analysis of gates with multiple inputs. Simulation results prove that the proposed F-SSTA algorithm has an error lower than 10 % compared to Monte-Carlo simulations. Furthermore, the evaluation time is more than 20 times faster compared to an accurate SSTA algorithm.

References

[1] S. Borkar, T. Karnik, and V. De, Design and reliability challenges in nanometer technologies, *41st Design Automation Conference*, USA, p. 75, 2004.

[2] Boning, D., and Nassif, S., Models of Process Variations in Device and Interconnect, in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan (ed.), 2000.

[3] M. Orshansky and K. Keutzer**,** A general probabilistic framework for worst-case timing analysis, *39th Design Automation Conference,* USA, pp. 556–561, 2002.

[4] A. Agarwal, F. Dartu, and D. Blaauw, Statistical Gate Delay Model Considering Multiple Input Switching, *41st Design Automation Conference*, USA, 2004.

[5] C. Visweswariah, Statistical Analysis and Design: From Picoseconds to Probabilities, *17. Symposium on Integrated Circuits and System Design*, Brazil, 2004.

[6] T. Sakurai and A. Newton, Alpha-Power Law MOSFET Model and its Application to CMOS Inverter Delay and other Formulas, In *IEEE Journal of Solid-State Circuits*, no. 2, pp. 584-5941990.

[7] F. Sill, and D. Timmermann, Total Leakage Reduction by Observance of Parameter Variations, *NORCHIP'05*, 2005.

[8] N.H.E. Weste and D. Harris, *CMOS VLSI Design – A Circuit and Systems Perspective*, 3rd edition, Addison Wesley, Boston, 2005.

[9] S.H. Choi, B.C. Paul, and K. Roy, "Novel Sizing for Yield Improvement under Process Variation in Nanometer Technology", *41st Design Automation Conference*, USA, pp. 454-459, 2004.

[10] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu, New paradigm of predictive MOSFET and interconnect modeling for early circuit design, Custom Integrated Circuits Conference, pp. 201-204, 2000