SSTA Framework Based on Moments Propagation

Zeqin Wu

University of Montpellier II
Science and Technology of Languedoc

Thesis

Doctor of Philosophy

Discipline : Electronics, Optronics and Systems
Doctoral School : Information, Structure and Systems
Doctoral Education : Automatic Systems and Microelectronics

WU Zeqin

December 11, 2009

SSTA Framework Based on Moments Propagation

Committee

ROBERT Michel	Professor	President
PIGUET Christian	Research Director CSEM	Reporter
BELLEVILLE Marc	Research Director CEA	Reporter
WILSON Robin	Research Director ST	Examinator
AMARA Amara	Professor	Examinator
MAURINE Philippe	Associate Professor	Examinator
DUCHARME Gilles	Professor	Supervisor
AZEMARD Nadine	Researcher CNRS	Supervisor
MAS André	Professor	Invitee

Table of Contents

List of Figures

List of Tables

PREFACE

As technology enters the nanometer era, the traditional ***Corner-based Timing Analysis*** (CTA) is predicted to no longer fully address the needs of IC designers in the near future. This prediction has urged the rapid development of ***Statistical Static Timing Analysis*** (SSTA). Since 2003, thousands of papers have been published in this field. However, SSTA is still in the very beginning state and much work needs to be done to improve it. Our research is on this front topic.

This thesis is organized into six chapters. The first chapter defines the problem of timing verification and discusses the need of SSTA. CHAPTER 2 focuses on the present state of SSTA. In CHAPTER 3, we introduce our path-based SSTA framework. CHAPTER 4 presents an improved method for timing characterization, which is a step to collect data to feed our SSTA engine. In CHAPTER 5, we apply the proposed SSTA framework and compare its results with those of CTA. Finally, CHAPTER 6 gives the conclusions and future work.

I would like to thank Philippe MAURINE and Nadine AZEMARD for providing me with the opportunity to do this research and their helps during these years. I also would like to acknowledge Gilles DUCHARME for his comments and corrections all along the redaction of this thesis. I also thank all my colleagues of LIRMM.

WU Zeqin

October 2009

CHAPTER

1

INTRODUCTION

This chapter first introduces the notions of propagation delay and timing constraint. Then, the problem of timing verification is defined. SECTION 1.2 presents briefly the traditional ***Corner-based Timing Analysis*** *(CTA), which has been widely used for timing verification in the past twenty years. In SECTION 1.3, we analyze the pessimism of CTA and conclude that this pessimism is increasing as the feature sizes are shrinking, which results in the need of* ***Statistical Static Timing Analysis*** *(SSTA). Finally, the outline of the thesis is given.*

Continuing advances in design techniques and fabrication process technology are leading to the design and manufacturing of very high performance ***Integrated Circuits*** (IC), i.e. high speed and low power consumption IC. The ever higher demand of performance from consumers allows for less and less design margins. In consequence, the propagation delay of an IC needs to be checked against increasingly tighter timing constraints that are related to the expected performance. At the same time, with the rapid decrease of minimum feature sizes, the effects of fabrication process fluctuations on timing characteristics are becoming significant. As a result, the traditional ***Corner-based Timing Analysis***[1] (CTA) is predicted to no longer fully address the needs of IC designers in the near future.

CTA has been a simple and efficient method for the timing verification of modern IC designs. By describing process and environmental variations with corners, gate-level delays (the basic primitives) of IC turn into deterministic quantities, and therefore are easy to be propagated. In micronic technologies, process variations are relatively small compared to supply voltage and temperature variations, so that modeling variations with extreme values produces acceptable outcomes. However, as technology enters the nanometer era (< 90 nm), it becomes difficult to construct guaranteed bounds on the circuit delay probability distribution without being overly conservative. Such pessimism of corner-based design methodologies leads to an increase in design effort, or a reduction of the relative timing performance to previous generation levels [1]. As a consequence, ***Statistical Static Timing Analysis*** (SSTA), which is considered as the replacement of CTA, has been developed and received considerable attention in the domain of ***Computer-Aided-Design*** (CAD) in the last few years. Rather than simply determine corners and attempt to arrive at a single value for delays, statistical timing engines propagate probability distributions. This statistical technique is more reasonable than CTA in nature and offers much more accurate estimation of actual circuit performance. Recent works [1], [2] claim that SSTA is absolutely necessary for future IC design.

[1] Also called ***Static Timing Analysis*** (STA). In this thesis, we use CTA in order to avoid confusion with another abbreviation: ***Statistical Static Timing Analysis*** (SSTA).

1.1 TIMING VERIFICATION

A successful digital IC design must provide the intended functionality and operate at the speed defined in the design requirements. Manufactured circuits that do not meet the specified timing constraints may be functionally incorrect and hence cannot be sold, or have to give up the market-related design goal by slowing down the speed. Consequently, a designer must perform timing verifications at numerous development steps before fabrication.

The essential objective of timing verification is to guarantee that circuit propagation delay satisfies the timing constraints given by the specifications. This is done by identifying the critical paths of a circuit, i.e. those paths that have the maximum delay. This information about critical paths can be used to decrease circuit delay, which is necessary if some timing constraints are violated, and is required to increase the clock frequency during design optimization. In addition, there are timing-verifiers that downsize high-speed gates along non-critical paths in order to save power consumption.

In the process of timing verification, the most crucial task is to estimate propagation delay, which is greatly affected by two sources of variations. The first source comprises environmental variations, such as supply voltage and temperature dispersions that arise during circuit operation. The second source comes from process variations due to manufacturing dispersions. In order to tolerate these variations, the timing behaviors of circuits need to be checked against the timing constraints under all possible combinations of environmental and process characteristics.

1.1.1 Propagation delay

In physics, ***propagation delay*** is the amount of time for a signal to travel to its destination. In digital circuits, it is usually defined as the interval between the time when the input waveform crosses the 50% point of its maximum supply voltage value V_{dd}, and the time when the corresponding output waveform crosses the same threshold. The ***transition time*** (or ***slope***) of a waveform is the time needed to switch from one stable state to another, such as from 0 to V_{dd} or the contrary. To avoid the effects of noises, especially those appearing at the head and the tail of a waveform, this transition time is defined as the time spent by the signal to go from *x*% to *y*% of

V_{dd}. In this thesis, all slopes are measured using the 20% – 80% specification. FIGURE 1.1 illustrates the definitions of propagation delay and slope. Note that a waveform can be classified into one of the two types:

- ***rising edge*** is the transition of a digital signal from 0 to V_{dd};
- ***falling edge*** is the V_{dd} to 0 transition.

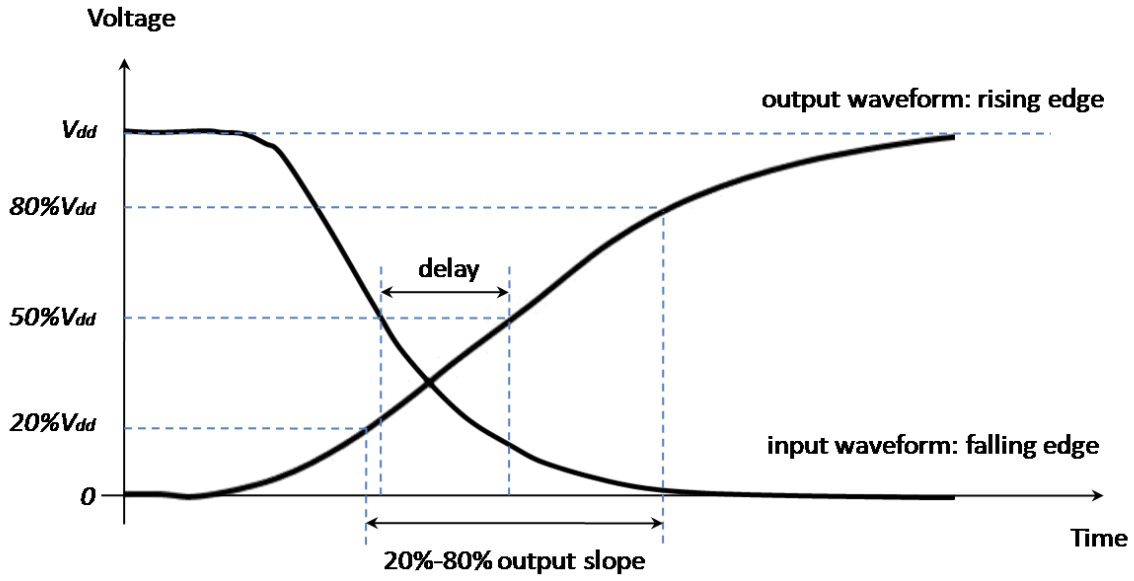


FIGURE 1.1 Illustration of propagation delay and slope

A digital IC consists of millions of transistors, organized into logic gates. Thus, propagation delay through a logic gate, called ***gate delay***, is the fundamental element for timing verification. The factors that affect gate delay include:

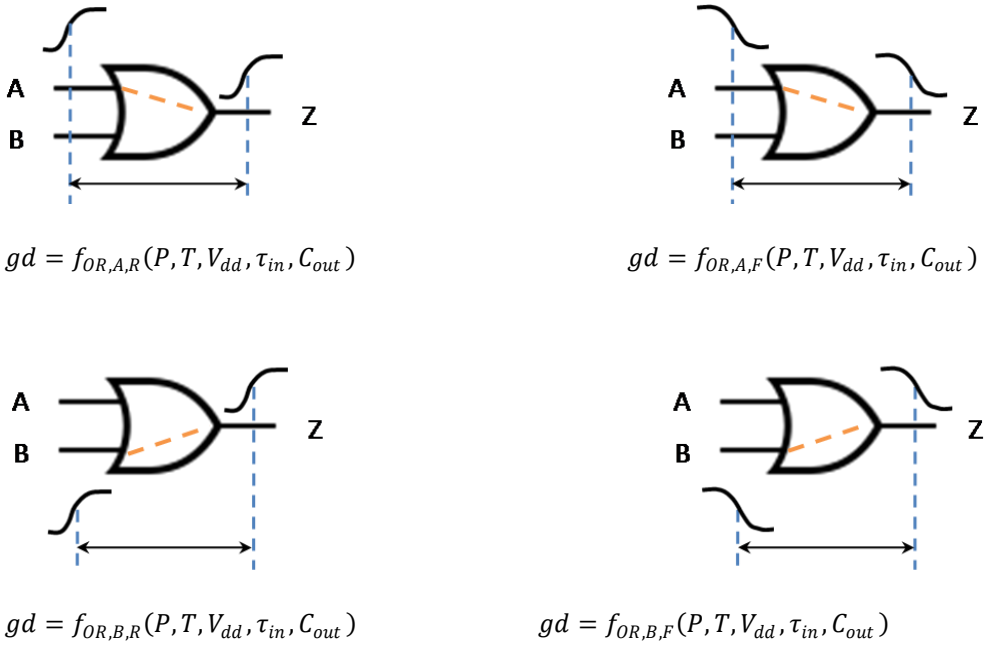
a) gate type ($INV, AND, OR, \ldots$), input pin ($A, B, \ldots$), and output edge (either rising edge or falling edge, abbreviated respectively to R, F);
b) process parameters $P = (p_1, p_2, \ldots, p_L)$, where $p_l, (l = 1, 2, \ldots, L)$ represent physical parameters, such as effective channel length L_{eff}, oxide thickness t_{ox}, etc;
c) environmental parameters: temperature T and supply voltage V_{dd};
d) operating conditions: input slope τ_{in} and output load[2] C_{out}.

[2] ***Load*** indicates all objects that are connected to the output of a gate: a capacitor, a resistor, a mixture of them, etc.

In general, gate delay is a complicated nonlinear function of the above factors, especially in the case where two or more inputs of a multiple-input gate switch simultaneously. To make gate delay modeling possible, it is necessary to set the assumption that only one input switches at any time for a multiple-input gate. Under such an assumption, given the gate type, input pin and output edge, the pin-to-pin gate delay can be modeled by:

$$gd = f_{type,pin,edge}(P, T, V_{dd}, \tau_{in}, C_{out}) \tag{1.1}$$

where $f_{type,pin,edge}$ is a function specific to gate type, input pin and output edge. As an example, for the two-input OR gate in FIGURE 1.2, there are four possible functions. Hence, under the single switching input assumption, the gate delay will take one of the four values outputted by the following functions $\{f_{OR,A,R}, f_{OR,A,F}, f_{OR,B,R}, f_{OR,B,F}\}$ according to gate type, input pin, and output edge.



$gd = f_{OR,A,R}(P, T, V_{dd}, \tau_{in}, C_{out})$

$gd = f_{OR,A,F}(P, T, V_{dd}, \tau_{in}, C_{out})$

$gd = f_{OR,B,R}(P, T, V_{dd}, \tau_{in}, C_{out})$

$gd = f_{OR,B,F}(P, T, V_{dd}, \tau_{in}, C_{out})$

FIGURE 1.2 Pin-to-pin gate delays of a two-input *OR* gate

Note that in the rest of this thesis, the delay of gate k will be denoted by gd_k for simplicity. This implies that the indices of the function $f_{type,pin,edge}$ and all its parameters are known from the context.

With the above gate delay definition, propagation delay can be extended to circuit-level. Consider a combinational circuit block which is composed of K gates and has I input pins $A_i, (i = 1, 2, \dots, I)$ and J output pins $Z_j, (j = 1, 2, \dots, J)$. As defined in SECTION 1.1.1, a transition is a change of states. Therefore, we may define $\mathbb{\Gamma}$ as the set of all possible transitions at all the input pins $A_i, (i = 1, 2, \dots, I)$ of the circuit. But only a subset Γ_{A_i,Z_j} of $\mathbb{\Gamma}$ produces an effective signal propagation[3] from the input pin A_i to the output pin Z_j.

For $\gamma_{in} \in \Gamma_{A_i,Z_j}$, we can first calculate all gate delays $gd_k, (k = 1, 2, \dots, K)$ considering the context of operation, i.e. the related $P_k, T_k, V_{dd,k}, \tau_{in,k}, C_{out,k}, (k = 1, 2, \dots, K)$ are known for each gate; Next, the circuit delay $cd_{A_i,Z_j,\gamma_{in}}$ from the input pin A_i to the output pin Z_j is computed by:

$$cd_{A_i,Z_j,\gamma_{in}} = h(gd_1, gd_2, \dots, gd_K) \tag{1.2}$$

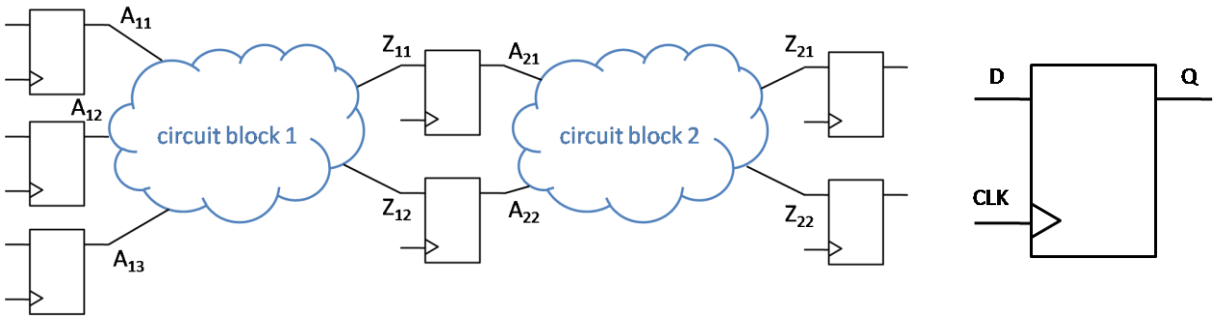
The function h in EQUATION (1.2) is simple, and involves only the essential operations SUM and MAX/MIN. However, since timing verification has entered the statistical era, estimation of $cd_{A_i,Z_j,\gamma in}$ has become a challenging task due to the fact that the MAX/MIN of random variables is difficult to determine.

1.1.2 Timing constraints

In the present-day field of microelectronics, almost all digital ICs can be simply described as a set of flip-flops that link different circuit blocks together. FIGURE 1.3(a) shows a diagram, in which a cloud represents a circuit block made of logic gates, while flip-flops are used to synchronize actions of circuit blocks with the help of a global clock signal. In FIGURE 1.3(a), considering propagation delay, it is rare that the output data of Z_{11} and Z_{12}, which is required respectively by A_{21} and A_{22}, arrives at the same moment. With flip-flops and an active clock edge used as control signal, difference in propagation delays is eliminated and the needed values are transferred simultaneously to the corresponding input A_{21} and A_{22} of the following circuit block.

[3] The propagation delay is non-null.

A simplified flip-flop is shown in FIGURE 1.3(b) and consists of a data input *D*, a clock input *CLK*, and an output *Q* which always takes on the state of the input *D* when the active clock edge is switching. However, such synchronous scheme is prone to the following meta-stability problem that happens when a data is changing at the instant of an active clock edge: the output may behave unpredictably, take much more time to settle to its correct state, or even oscillate several times before settling. This problem can be avoided by ensuring that the data is held valid and constant for specified period before and after the clock rising edge, called the setup time and the hold time respectively. The ***setup time*** is the minimum time before the arrival of an active clock edge during which the input data must be valid for reliable latching. Similarly, the ***hold time*** represents the minimum time during which the data input must be held stable after the active clock edge.



(a) Diagram of digital IC (b) Simplified flip-flop

FIGURE 1.3 Diagram of digital IC: a set of flip-flops linking circuit blocks

FIGURE 1.4 illustrates the setup time and hold time constraints with a simple block. If the clock period T_{CLK} is given, then for any input transition $\gamma_{in} \in \Gamma_{A_i,Z_j}$, the two timing constraints can be expressed mathematically by:

$$gd_{[CLK_0 \to CLK_{A_1}]} + gd_{[CLK_{A_1} \to A_1]} + cd_{A_1,Z_1,\gamma_{in}} < gd_{[CLK_0 \to CLK_{Z_1}]} - d_{setup} + T_{CLK} \quad (1.3)$$

$$gd_{[CLK_0 \to CLK_{A_1}]} + gd_{[CLK_{A_1} \to A_1]} + cd_{A_1,Z_1,\gamma_{in}} > gd_{[CLK_0 \to CLK_{Z_1}]} + d_{hold} \quad (1.4)$$

where $gd_{[X \to Y]}$ indicates any possible delay propagating from pin X to pin Y, and d_{setup}, d_{hold} are respectively the setup time and the hold time of the flip-flop FF_{Z_1}.

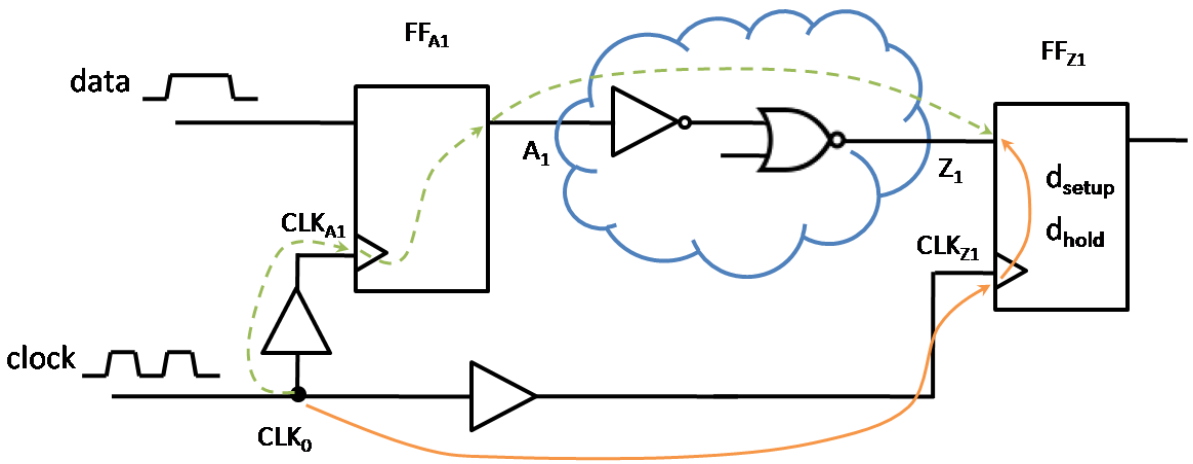
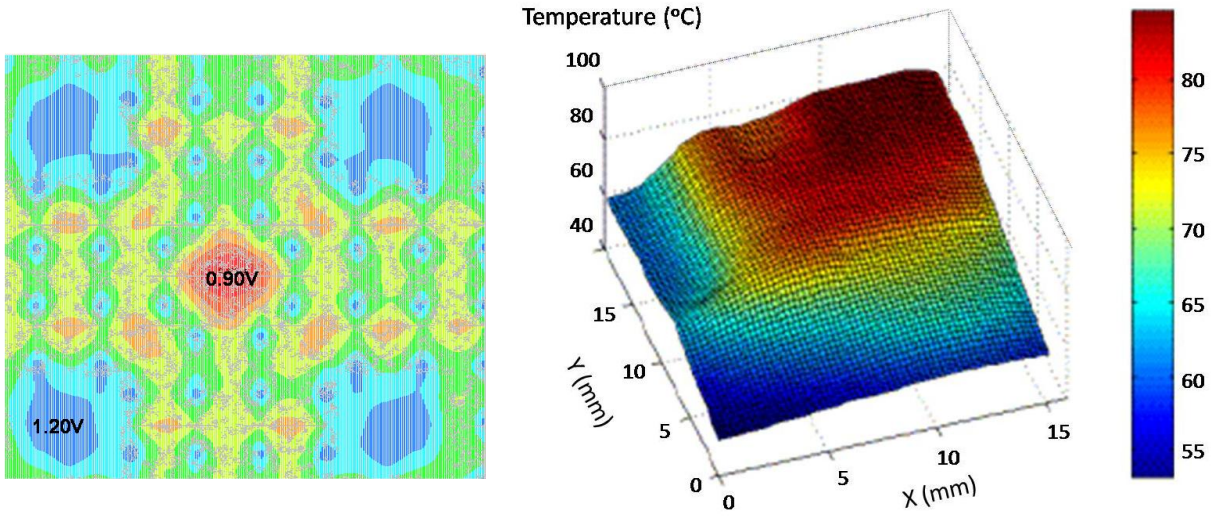


FIGURE 1.4 Setup time and hold time constraints of flip-flop FF_{Z_1}

Theoretically, if the setup time constraint (1.3) is violated, slowing down the clock will increase the clock period T_{CLK} and enable the right value to be latched. On the other hand, if a hold time violation problem occurs, it cannot be solved by giving up the design specifications and will lead to functional faults.

1.1.3 Source of variations

Among all the factors that affect propagation delay discussed in SECTION 1.1.1, gate type, input pin, and output edge are known and fixed; the others are variational. These variations are directly or indirectly caused by two types of sources. First, ***environmental variations***, as the name suggests, are variations of the surrounding environment in which a circuit sits during its operation. These variations include temperature variations and variations in supply voltage. FIGURE 1.5 gives an example of the environmental variations across an IC. The uneven supply voltage distribution and the spatial variations of temperature shown in FIGURE 1.5, come from the variation in switching activities. From the two panels of this figure, it is obvious that the components contained within the IC work under different supply voltage and temperature conditions. To avoid the loss of accuracy when estimating propagation delay, a reasonable model to describe and predict the environmental variations is required. But the modeling task is challenging because this category of variations is time-dependent.



(a) Variations in supply voltage

(b) Temperature variations

FIGURE 1.5 Environmental variations across an IC [3]

The second source of variations is ***process variations*** from perturbations in the fabrication process and physical limitations. These manufacturing variations cause deviations (from intended or designed values) of physical parameters and thus have significant impact on propagation delay. Unlike time-varying environmental variations, physical parameters are essentially permanent after the fabrication. However, during the design procedure, the randomness of some of these process variations must be taken into account. This randomness leads to the fact that propagation delays in EQUATIONS (1.1) – (1.2), are randomly distributed, which is the main difficulty in timing verification.

1.1.4 Mathematical description

Consider a simplified circuit: a combinational circuit block links respectively I identical flip-flops FF_{A_i} at input pins $A_i, (i = 1, 2, \dots, I)$ and J identical flip-flops FF_{Z_j} at output pins $Z_j, (j = 1, 2, \dots, J)$. Under the assumption that physical parameters $P = (p_1, p_2, \dots, p_L)$ are randomly distributed, all timing parameters in EQUATIONS (1.3) – (1.4) are random except for the clock period T_{CLK}. Thus, we define two random variables $SS_{A_i,Z_j,\gamma_{in}}$ and $HS_{A_i,Z_j,\gamma_{in}}$, called respectively ***Setup Slack*** and ***Hold Slack***, as:

$$SS_{A_i,Z_j,\gamma_{in}} \stackrel{\text{def}}{=} \left\{ gd_{\left[CLK_0 \to CLK_{A_i}\right]} + gd_{\left[CLK_{A_i} \to A_i\right]} + cd_{A_i,Z_j,\gamma_{in}} \right\} - \left\{ gd_{\left[CLK_0 \to CLK_{Z_j}\right]} - d_{setup} \right\} \quad (1.5)$$

$$HS_{A_i,Z_j,\gamma_{in}} \stackrel{\text{def}}{=} \left\{ gd_{\left[CLK_0 \to CLK_{A_i}\right]} + gd_{\left[CLK_{A_i} \to A_i\right]} + cd_{A_i,Z_j,\gamma_{in}} \right\} - \left\{ gd_{\left[CLK_0 \to CLK_{Z_j}\right]} + d_{hold} \right\} \quad (1.6)$$

where $\gamma_{in} \in \Gamma_{A_i,Z_j}$. We assume that the setup time d_{setup} of each flip-flop follows the same probability distribution, and so does the hold time d_{hold}. With $SS_{A_i,Z_j,\gamma_{in}}$ and $HS_{A_i,Z_j,\gamma_{in}}$, we rewrite the two timing constraints as:

$$SS_{A_i,Z_j,\gamma_{in}} < T_{CLK} \quad (1.7)$$

$$HS_{A_i,Z_j,\gamma_{in}} > 0 \quad (1.8)$$

Before defining the problem of timing verification, we further assume that:

a) supply voltage and temperature of each gate are respectively bounded by known values V_{min}, V_{max} and T_{min}, T_{max}, i.e. $V_{dd} \in [V_{min}, V_{max}]$ and $T \in [T_{min}, T_{max}]$;
b) the probability distribution F_l of each process parameter p_l, $(l = 1, 2, \ldots, L)$ is known;
c) for any two gates k and m, their process parameters $p_{l,k}$ and $p_{l,m}$ are dependent.

Given a clock signal, a clock period T_{CLK}, and a probability $\theta \in (0, 1)$, then $SS_{A_i,Z_j,\gamma_{in}}$ and $HS_{A_i,Z_j,\gamma_{in}}$ must satisfy the condition:

$$Pr\left\{ \bigcap_{i=1}^{I} \bigcap_{j=1}^{J} \bigcap_{\gamma_{in} \in \Gamma_{A_i,Z_j}} \left[\left(SS_{A_i,Z_j,\gamma_{in}} < T_{CLK} \right) \cap \left(HS_{A_i,Z_j,\gamma_{in}} > 0 \right) \right] \right\} \geq \theta \quad (1.9)$$

Note that the two timing constraints in EQUATIONS (1.7) – (1.8) are similar because they bound random variables. What is more, the setup time constraint can conversely be used to determine the initial clock signal and the appropriate clock period. Hence, in the rest of this thesis, we will mainly discuss the setup time problem.

1.2 CORNER-BASED TIMING ANALYSIS

Although theoretically, timing verification can be undertaken using electrical circuit simulation, such an approach is too slow to be practical. In the past decades, ***Corner-based Timing Analysis*** (CTA) offered quick and reasonably accurate estimations of propagation delays. This timing method assumes that the best-corners or worst-corners of the process and environmental parameters occurs simultaneously, and verifies timing behaviors under these extreme conditions. In other words, variations are replaced by deterministic quantities. The basic idea behind this approach is that if a circuit works correctly in extreme cases, then it will also work correctly under normal conditions.

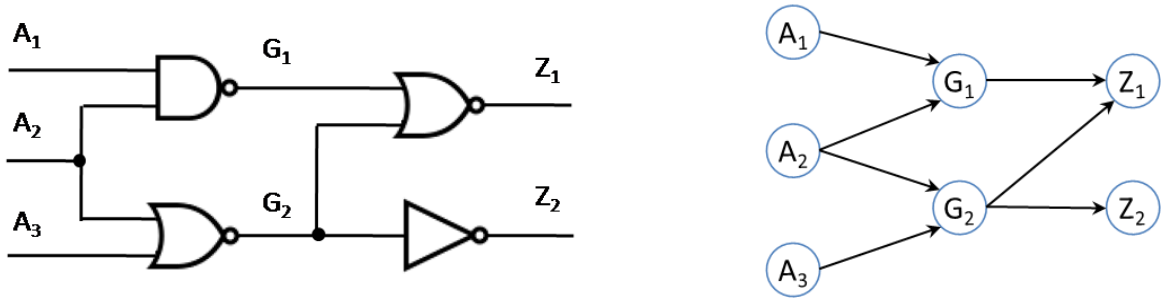
1.2.1 Basic concepts of timing analysis

A circuit may be represented as a ***timing graph*** $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ is a set of nodes, and $\mathbb{E}$ is a set of edges. A node $v_i \in \mathbb{V}$ corresponds to a net in the circuit. The edge $e_{v_i,v_j} \in \mathbb{E}$ represents the propagation delay between two adjacent nodes v_i and v_j. Each edge e_{v_i,v_j} has a pin-to-pin gate delay gd_{v_i,v_j} as the weight; and each node has a delay related term t_{v_i}, called ***arrival time***. Note that a timing graph is oriented from the primary inputs to the primary outputs of the corresponding circuit.

A simple combinational circuit and its corresponding timing graph (without considering interconnects) are illustrated respectively in FIGURE 1.6(a) and 1.6(b). Compared with the circuit diagram, an edge e_{v_i,v_j} corresponds to a pin-to-pin gate delay, and a node v_i is either a net, or a primary input pin, or a primary output pin.

Another useful term is ***timing path***. In the context of digital circuit, a timing path is a set of connected edges between an input node A_i and an output node Z_j, such as $\{e_{A_1,G_1}, e_{G_1,Z_1}\}$ and $\{e_{A_3,G_2}, e_{G_2,Z_1}\}$ in FIGURE 1.6(b). ***Path delay*** is the sum of weights of all edges on a timing path. Note that path delay is a little different from the pin-to-pin circuit delay defined in EQUATION (1.2): in FIGURE 1.6(b), the pin-to-pin circuit delay $cd_{A_2,Z_1,\gamma_{in}}$ may be one of the two path

delays: $gd_{A_2,G_1} + gd_{G_1,Z_1}$ and $gd_{A_2,G_2} + gd_{G_2,Z_1}$, each of which corresponds to an input transition γ_{in} applied at the input pins A_1, A_2 and A_3.



(a) A simple combinational circuit diagram (b) Corresponding timing graph

FIGURE 1.6 Illustration of timing graph

1.2.2 Modeling variations with corners

As stated before, the key idea of CTA is that randomly distributed process variables and time-dependent environmental parameters are replaced by fixed and deterministic corners. For the setup time check, these parameters are set at their worst values so that the maximum circuit delay can be computed. These corners of parameters can be identified according to a sensitivity analysis of the function $f_{type,pin,edge}$ in EQUATION (1.1).

The worst corners of supply voltage V_{dd} and temperature T are respectively V_{min} and T_{max}. In addition, from SECTION 1.1.4, the probability distributions F_l of $p_l, (l = 1, 2, \dots, L)$ are known, so that for a given probability $\beta \in (0, 1)$, the upper extreme bound $p_{upr,l}$ and the lower extreme bound $p_{lwr,l}$ of each process parameter can be derived by:

$$\begin{cases} 1 - F_l(p_{upr,l}) = Pr(p_l \geq p_{upr,l}) = \dfrac{\beta}{2} \\ F_l(p_{lwr,l}) = Pr(p_l \leq p_{lwr,l}) = \dfrac{\beta}{2} \end{cases} \tag{1.10}$$

We assume that for each process parameter p_l, the function $f_{type,pin,edge}$ is either monotone decreasing or monotone increasing. Without loss of generality, suppose that $f_{type,pin,edge}$ is

decreasing for each p_l, then the maximum gate delay can be obtained with V_{min}, T_{max} and $p_{lwr,l}, (l = 1, 2, \dots, L)$. In practice, the probability distribution F_l of p_l is assumed to be Gaussian, denoted as $p_l \sim N(\mu_{p_l}, \sigma_{p_l}^2)$, and the parameters $\mu_{p_l}, \sigma_{p_l}^2$ of these distributions are estimated by empirical data. In addition, β is usually set to 0.003, which gives the worst process corners $p_{lwr,l} = \mu_{p_l} - 3 \cdot \sigma_{p_l}$.

1.2.3 Estimation of circuit delay

From the discussion in SECTION 1.2.2, corners are set for the parameters P, V_{dd}, T in EQUATION (1.1). C_{out} is considered as a known constant, because the variations in C_{out} are small enough to be neglected. As regards τ_{in}, if P, V_{dd}, T are at their worst corners, it will also reach its worst corner value, guarantying the worst estimation of delay. Finally, we use lookup table and bilinear interpolation [4] techniques to approximate the complicated function $f_{type,pin,edge}$. A lookup table is generated with the help of numerical results from circuit simulation. Typically, for the worst combination V_{min}, T_{max} and $p_{lwr,l}, (l = 1, 2, \dots, L)$, the function in EQUATION (1.1) is reduced to a simple one depending only on τ_{in} and C_{out}. Thus, for any logic gate, applying a linear ramp signal of slope τ_{in} at one of the input pins and a capacitor of charge C_{out} at the output pin, the pin-to-pin worst gate delay is obtained by circuit simulation.

Having modeled gate delays with lookup tables, the next step is to estimate circuit-level delay and verify the timing constraint. The corner-based model permits us to rewrite condition (1.9) as:

$$Pr\left\{\left(\max_{\gamma_{in} \in \mathbb{\Gamma}^*}\left(SS_{A_i,Z_j,\gamma_{in}}\right) < T_{CLK}\right) \cap \left(\min_{\gamma_{in} \in \mathbb{\Gamma}^*}\left(HS_{A_i,Z_j,\gamma_{in}}\right) > 0\right)\right\} \geq \theta \tag{1.11}$$

where $\mathbb{\Gamma}^* = \left\{\bigcup_{i=1}^{I}\bigcup_{j=1}^{J} \Gamma_{A_i,Z_j}\right\}$ is a subset of $\mathbb{\Gamma}$, which is the set of all possible input transitions defined in SECTION 1.1.1. Combining EQUATIONS (1.5) – (1.6) with EQUATION (1.11), the setup time and the hold time checks can be respectively translated into the computation of $max_{\gamma_{in} \in \mathbb{\Gamma}^*}\left(cd_{A_i,Z_j,\gamma_{in}}\right)$ and $min_{\gamma_{in} \in \mathbb{\Gamma}^*}\left(cd_{A_i,Z_j,\gamma_{in}}\right)$. For this purpose, we convert the timing graph in FIGURE 1.6(b) into one that has a single source node I and a single sink node O. This converted timing graph is shown in FIGURE 1.7. After this slight modification, the timing verification problem can be solved using ***Performance Evaluation and Review Technique*** (PERT) [5]

of operational research. As an example, for the setup time check, the arrival time t_{G_1} of node G_1 is given by:

$$t_{G_1} = max\left(t_{A_1} + gd_{A_1,G_1}, t_{A_2} + gd_{A_2,G_1}\right) \tag{1.12}$$

Here gd_{A_1,G_1} and gd_{A_2,G_1} represent the pin-to-pin gate delays. If we apply iteratively EQUATION (1.12) for each node in the graph, the maximum circuit delay can be easily computed.

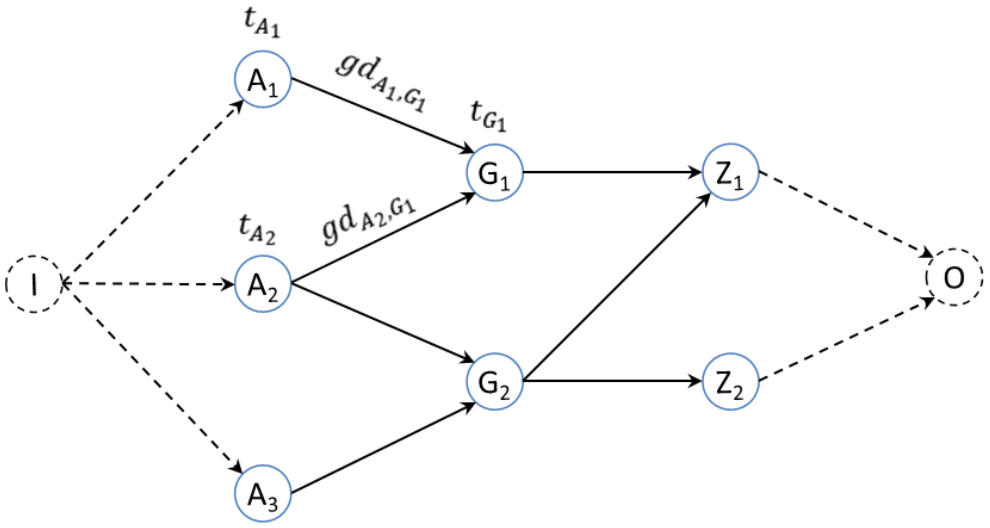


FIGURE 1.7 A PERT task graph

1.3 ON THE NEED OF STATISTICAL STATIC TIMING ANALYSIS

CTA assumes that all physical and environmental parameters are at their worst or best conditions simultaneously. From the point of view of probability theory, this conservative case is next to impossible to appear in reality. Consequently, such an assumption induces pessimism in delay estimation, and thereby in circuit design. As the magnitude of process variations grows, this pessimism increases significantly, leading to the understanding that traditional corner-based design methodologies will not meet the needs of designers in the near future. Therefore, ***Statistical Static Timing Analysis*** (SSTA), where process variations and timing characteristics are considered as random variables, has gained favor in the past six years. By propagating delay

probability distributions through a circuit instead of pessimistic delay quantities, we may arrive at a much more accurate estimate of circuit delay.

1.3.1 Increasing pessimism of corner-based methods

As feature sizes continue to shrink, process variations σ_{p_l} are increasing relative to their means μ_{p_l}. FIGURE 1.8 shows the increase in the variability of key process parameters, such as oxide thickness t_{ox} and transistor width W. As an example, the proportion of variations in gate-length L_{eff} to its corresponding mean has increased from 35% in a 130 nm technology to almost 60% in a 65 nm technology. Besides, these increasing variations must be coupled with the fact that the number of process parameters whose variability must be taken into account has exploded in the past years. Due to these trends, some weaknesses of corner-based methods are becoming obvious.

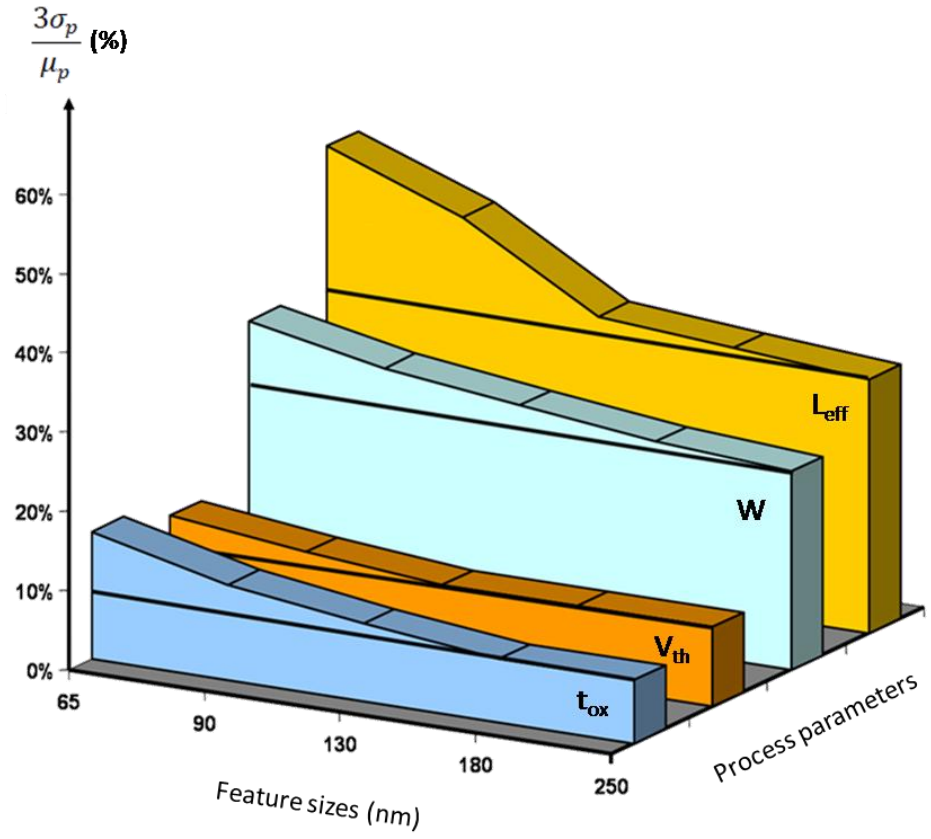


FIGURE 1.8 Variability trends in key process parameters with shrinking feature sizes [6]

To illustrate the weakness of replacing random process variations with corners, we consider a simplified case where the propagation delay gd of an inverter is a sum function of all the process parameters:

$$gd = \sum_{l=1}^{L} p_l \tag{1.13}$$

Here, $p_l, (l = 1,2, \dots, L)$ are assumed Gaussian distributed with mean μ_{p_l} and variance $\sigma_{p_l}^2$. Besides, for any $l_1 \neq l_2$, we suppose the correlation $cor(p_{l_1}, p_{l_2}) = 0$ and $\mu_{p_{l_1}} = \mu_{p_{l_2}}, \sigma_{p_{l_1}} = \sigma_{p_{l_2}}$. Note that p_{l_1}, p_{l_2} are two different parameters of the same gate while $p_{l,k}, p_{l,m}$ $(k \neq m)$ indicate parameters of the same type for two different gates.

Then the probability distribution of gate delay $gd \sim N(\mu_{gd}, \sigma_{gd}^2)$ is computed by:

$$\begin{cases} \mu_{gd} = \sum_{l=1}^{L} \mu_{p_l} = L \cdot \mu_{p_1} \\ \sigma_{gd} = \sqrt{\sum_{l=1}^{L} \sigma_{p_l}^2} = \sqrt{L} \cdot \sigma_{p_1} \end{cases} \tag{1.14}$$

The worst gate delay w_{gd} is computed by CTA as:

$$w_{gd} = \sum_{l=1}^{L} (\mu_{p_l} + 3 \cdot \sigma_{p_l}) = L \cdot \mu_{p_1} + 3L \cdot \sigma_{p_1} \tag{1.15}$$

Comparing the worst gate delay w_{gd} and the statistical 3σ corner of gate delay yields:

$$\omega = \frac{w_{gd} - (\mu_{gd} + 3 \cdot \sigma_{gd})}{\mu_{gd}} = \frac{3(L - \sqrt{L}) \cdot \sigma_{p_1}}{L \cdot \mu_{p_1}} = 3(1 - L^{-0.5}) \cdot \frac{\sigma_{p_1}}{\mu_{p_1}} \tag{1.16}$$

If $L = 3$ and $\sigma_{p_1}/\mu_{p_1} = 0.15$, then the normalized rate ω is about 0.2, indicating that the overestimate of worst gate delay is 20% of the delay mean. As shown in FIGURE 1.8, for any p_l, the ratio σ_{p_l}/μ_{p_l} increases with each generation of technology, which results in the increase of the rate ω.

Note also that the pessimism of w_{gd} becomes more serious if the number of process parameters L is larger. This is the case in reality. As an example, the BSIM *v3* model has about $L = 50$ random process parameters, whereas the *v4* version needs $L = 80$ parameters or so [7]. If ω_{v3} and ω_{v4} represent the rates of these two BSIM models and have the same ratio σ_{p_1}/μ_{p_1}, then according to EQUATION (1.16), we have $(\omega_{v4} - \omega_{v3})/\omega_{v3} \approx 0.03$, which means that the pessimism will increase 3% if the inverter above is modeled by the BSIM *v4* instead of the *v3* version. Mathematically, according to EQUATIONS (1.14) – (1.15), we have:

$$\lim_{L\to+\infty} Pr(gd > w_{gd}) = \lim_{L\to+\infty} Pr\left(gd > \left(\mu_{gd} + 3\sqrt{L}\cdot\sigma_{gd}\right)\right) = 0 \tag{1.17}$$

which implies that the probability of gate delay exceeding the worst delay converges to zero if the number of parameters L increases. In other words, w_{gd} is too pessimistic.

Another weakness of CTA comes from gate-to-gate delay correlation. To see this more clearly, set the number of process parameters to $L = 1$, and combine EQUATION (1.14) with (1.15):

$$w_{gd} = \mu_{p_1} + 3\cdot\sigma_{p_1} = \mu_{gd} + 3\cdot\sigma_{gd} \tag{1.18}$$

Then a path with K gates has the worst path delay w_{pd} given by:

$$w_{pd} = \sum_{k=1}^{K} w_{gd_k} = \sum_{k=1}^{K}\left(\mu_{gd_k} + 3\cdot\sigma_{gd_k}\right) = \sum_{k=1}^{K}\mu_{gd_k} + 3\cdot\sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} 1\cdot\sigma_{gd_k}\sigma_{gd_m}} \tag{1.19}$$

As well, we estimate the statistical 3σ corner of path delay by:

$$\mu_{pd} + 3\cdot\sigma_{pd} = \sum_{k=1}^{K}\mu_{gd_k} + 3\cdot\sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K}\rho_{km}\cdot\sigma_{gd_k}\sigma_{gd_m}} \tag{1.20}$$

where pd is the path delay following the Gaussian distribution $pd \sim N\left(\mu_{pd}, \sigma_{pd}^2\right)$ and ρ_{km} is the correlation between gd_k and gd_m, i.e. $\rho_{km} = cor(gd_k, gd_m)$. Comparing EQUATION (1.19) with (1.20), we can find that the value "1" in EQUATION (1.19) corresponds to the gate-to-gate delay correlation ρ_{km} in EQUATION (1.20). As we know $\rho_{km} \in [-1,1]$, w_{pd} is therefore over-estimating by setting the correlation $cor(gd_k, gd_m)$ to its maximal value "1". Similarly, the

circuit-level correlation or the path-to-path delay correlation, especially those in EQUATIONS (1.5) – (1.6), (1.9) are also estimated conservatively, either by "1" or "−1".

From the discussion above, the pessimism of CTA results becomes more problematic when:

a) the ratio of process variations to their nominal values is higher;
b) the number of process parameters L is larger;
c) the true correlation between delays is not close to either "1" or "−1".

1.3.2 SSTA moving from interesting to necessary

When process variations were relatively small compared to supply voltage and temperature variations, working with corners produced acceptable outcomes. However, the increasing variability in the manufacturing process and the ever tighter timing constraints lead to more and more efforts when designing circuit with corner-based methodologies.

FIGURE 1.9 illustrates the increasing pessimism of CTA and the tightening timing constraints. As shown in this figure, if the feature size decreases from 130 nm to 65 nm, i.e. nominal values of process parameters decrease, then the propagation delay will reach a lower level, which allows us to design ICs with tighter timing constraints (smaller clock periods $T_{CLK_2} < T_{CLK_1}$). At the same time, as discussed in SECTION 1.3.1, the results of CTA at 65 nm are more pessimistic than those at 130 nm. In FIGURE 1.9, w_1, w_2 denote the worst delays, and the statistical 3σ corner of delay distributions are:

$$s_i = \mu_i + 3\sigma_i \qquad (i = 1, 2) \tag{1.21}$$

where μ_i and σ_i are the corresponding delay mean and standard deviation. Then, the increasing pessimism leads to:

$$w_2 - s_2 > w_1 - s_1 \tag{1.22}$$

In consequence, the ***timing margin***, defined as $T_{CLK} - w$, gets smaller with each generation of technology. It is predicted that, in the near future, worst delays estimated by CTA could not be bounded by defined clock periods, i.e. we could not design an IC to satisfy the timing constraints using corner-based CAD tools. Such an outlook has resulted in a rapid development of SSTA in recent years.

There is no doubt that SSTA is a leading-edge technology. As the new promising generation of timing analysis, SSTA attacks the limitations of CTA by modeling process variations with probability distributions. Even though the accuracy of SSTA approaches is not fully clear yet, some statistical CAD tools have appeared and are already being used in the industry.

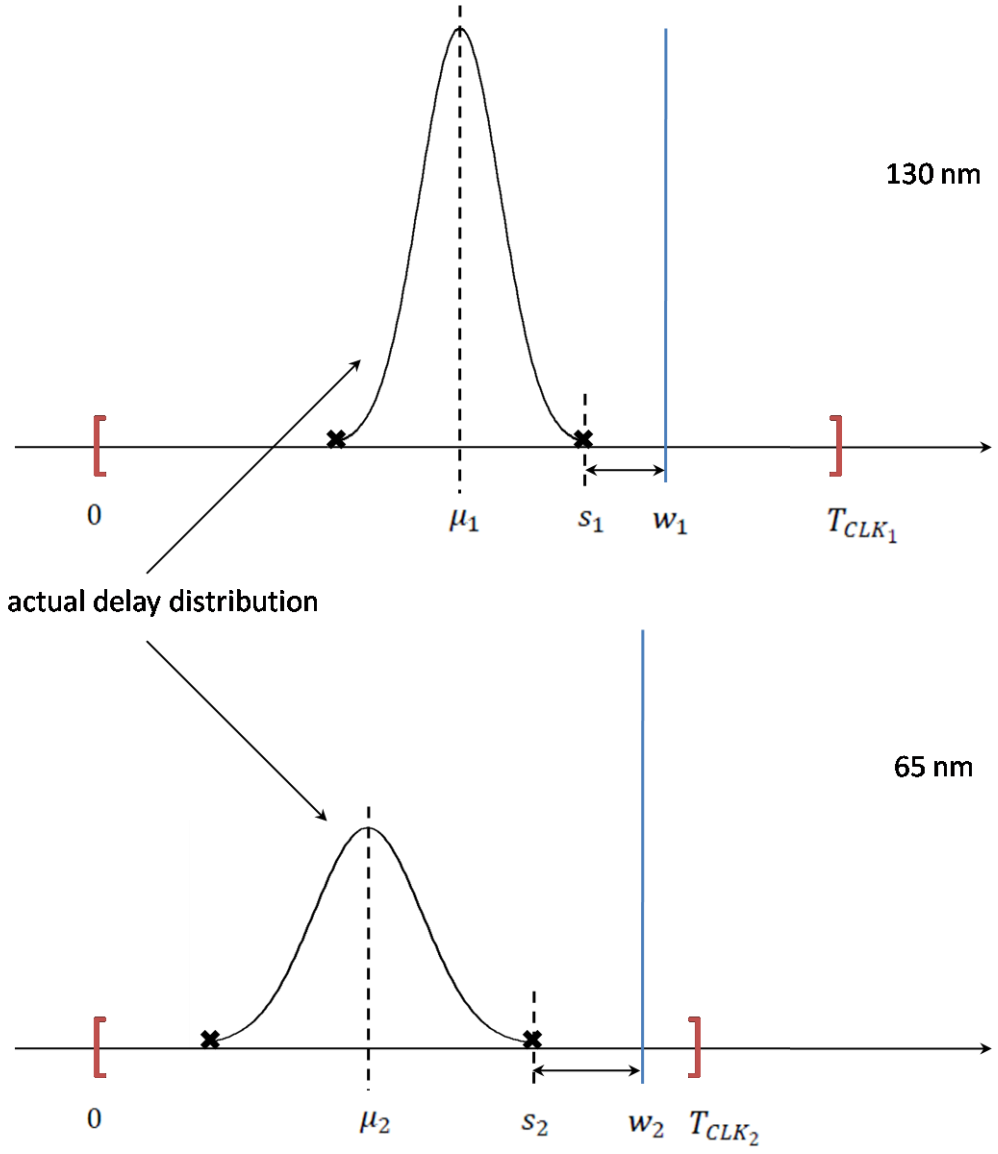


FIGURE 1.9 Increasing pessimism of CTA and tightening timing constraints

The authors of [2] believe that designs at 90 nm can benefit from the application of SSTA. But many industry experts feel that SSTA will not see widespread adoption until the 45 nm node becomes prevalent. [1] argues that SSTA is just about a must at 45nm, and definitely necessary at 32nm. To date, most designers see traditional CTA and SSTA as complementary.

Traditional CTA required over a decade to move from academic proposal to broad industry adoption. As well, algorithms for IC design based on statistical descriptions of process variations will probably take a decade to achieve meaningful industrial usage. It remains to be seen how long the process of widespread industrial adoption will take for SSTA. In addition to research on improved and enhanced SSTA, researchers are increasingly turning their attention to optimization of circuit design with the help of statistical techniques.

1.4 OUTLINE OF THE THESIS

The previous sections give answers to the following three questions: What is the role of timing analysis in the IC design flow? What is CTA? Why SSTA is becoming necessary?

CHAPTER 2 focuses on the present state of SSTA, including: the classification of SSTA methods, an overview of existing statistical timing techniques and their weaknesses, and the outlook of SSTA.

In CHAPTER 3, we introduce our path-based SSTA framework. With the help of conditional moments, the proposed SSTA engine computes path delays by propagating iteratively mean and variance of gate delay, which allows taking into account effects of input slope and output load. Moreover, we propose a technique to estimate cell-to-cell delay correlation. This chapter closes with a validation and a discussion of the framework.

In CHAPTER 4, we improve the conventional method of doing timing characterization, which is a step to collect data to feed the SSTA engine. The improvements include a Log-Logistic distribution based input signal and a technique to capture output load variations. Another concerning problem – acceleration of characterization, is addressed in this chapter as well.

In CHAPTER 5, we apply the SSTA framework and compare its results with those of CTA. First, some comparisons are given to show the gain of SSTA. Next, the discrepancy between orderings of critical paths obtained respectively by SSTA and by CTA is interpreted. Finally, we study the factors that affect cell-to-cell delay correlation for optimization of circuit design.

Finally, CHAPTER 6 gives the conclusions and future work.

CHAPTER

2

SSTA: STATE OF THE ART

This chapter provides an overview of the current state of ***Statistical Static Timing Analysis*** *(SSTA). Most of existing SSTA can be classified into parametric and Monte Carlo methods.* SECTION *2.1 summarizes these two categories of methods, and compares their advantages and disadvantages. In* SECTION *2.2, some widely adopted models and techniques are presented. In* SECTION *2.3, we discuss the common weaknesses of existing techniques and the outlook for SSTA.*

In recent years, the ever increasing variations of process parameters have raised concerns over the ability of ***Corners-based Timing Analysis*** (CTA) to accurately estimate circuit performance. It is now common belief that traditional deterministic ***Computer-Aided-Design*** (CAD) tools will not meet the needs of circuit designers in the future. As a result, ***Statistical Static Timing Analysis*** (SSTA), which is considered as a promising alternative, has developed greatly. Many companies now feel that the levels of variability are so high that the day of statistical CAD has arrived.

2.1 REVIEW OF SSTA

Some of the initial research works of SSTA date back to the introduction of timing analysis in the 1960s [8] as well as the early 1990s [9], [10]. However, the vast majority of research works on SSTA date from 2001, with thousands of papers published in this field in the last six years.

Most of the existing SSTA methods can be classified into two categories: parametric and Monte Carlo methods. ***Parametric methods*** [10] – [23] model process variations with random variables, and translate these variations to gate delays and arrival times through approximating polynomial models. These methods typically propagate arrival times through the timing graph by performing SUM and MAX/MIN operations. In contrast, ***Monte Carlo methods*** [24] – [27] employ complicated electrical models, fed by random inputs, to accurately reflect timing behaviors. This is feasible because circuit component behaviors obey to deterministic electrical laws whose parameters follow probability distributions.

2.1.1 Parametric methods

According to the algorithm to explore timing graphs, the existing parametric methods fall into one of the two categories shown in FIGURE 2.1: block-based algorithm [11] – [20] and path-based algorithm [10], [21] – [23]. A ***block-based algorithm*** performs a topological PERT-like (***Performance Evaluation and Review Technique)*** traversal of the timing graph. Compared with the CTA algorithm presented in SECTION 1.2.3, the only difference is that gate delays and arrival times are replaced by statistical distributions instead of being deterministic quantities. The

arrival time at each node is computed using two basic operations:

a) for all input edges of a particular node, the edge delay is convoluted (statistical SUM operation) with the arrival time at the source node of the edge;
b) given these resulting arrival time distributions, the final arrival time distribution at the node is estimated using approximated MAX operations.

The computation of the SUM operation is not difficult; however, finding the statistical MAX of two correlated arrival times is not trivial.

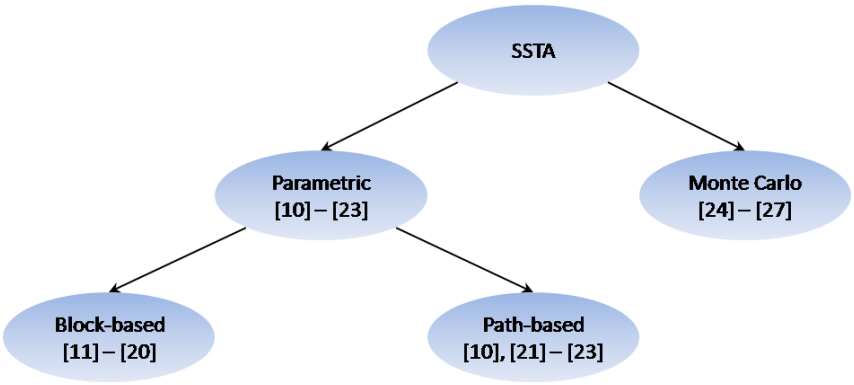


FIGURE 2.1 Classifications of existing SSTA methods

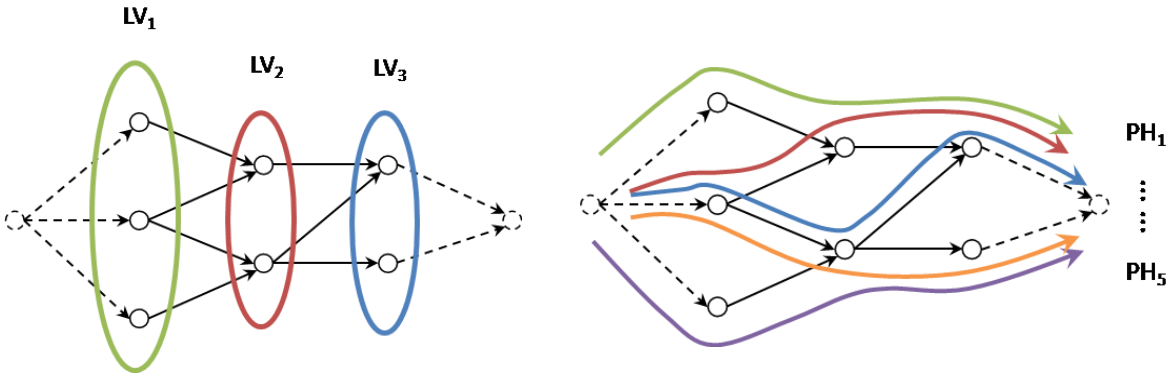
The key advantage of a block-based SSTA method is that the runtime is linear with circuit size [11] – [13]. Due to this competitive advantage, the block-based algorithm has been used in many current researches. Furthermore, a block-based method lends itself to incremental analysis, which is advantageous for optimization applications [13]. On the negative side, block-based methods suffer from a lack of accuracy especially for the approximated MAX operation [28].

In a ***path-based algorithm***, a set of paths, which are likely to become critical, is identified, and the delay distribution of each path is computed by convoluting (i.e. summing) the delay distributions of all its edges. Finally, the circuit delay distribution is computed by performing a statistical MAX operation over all the path delays.

The main advantage of this algorithm is that the analysis is split into two parts: the computation of each path delay distribution followed by the statistical MAX operation over these distributions [29]. Hence, much of the initial research in SSTA pertained to path-based algorithm. On the

negative side, the difficulty of the algorithm is in finding the above set of candidate paths so that no path with significant probability of being critical is excluded [29].

These two parametric statistical timing algorithms differ in accuracy and computational cost [28]. The path-based algorithm is simple and relatively accurate while the block-based algorithm considers the whole circuit and is of low computational cost. In FIGURE 2.2, we compare these two algorithms using the timing graph shown in FIGURE 1.7. FIGURE 2.2(a) illustrates the necessary levels to complete the topological traversal, and FIGURE 2.2(b) shows the five possible timing paths.



(a) Levels (LV_i) of the block-based algorithm (b) Set of paths (PH_i) for the path-based algorithm

FIGURE 2.2 Illustration of SSTA algorithms

It should be stated that the computational costs of parametric methods are far lower than those of Monte Carlo methods discussed in the next section. This is the only, but decisive, advantage of parametric methods. However, for broader adoption, the weaknesses of the current parametric SSTA should be overcome. According to [29], the main drawback is that they are based on models, where some of the timing and process variation effects are ignored or simplified, such as:

a) nonlinearity of gate delays as a function of the process parameters, input slope and output load;
b) approximations of the MAX operation;
c) interdependency among input/output edges and gate delay;
d) assumptions about probability distributions of process variations;
e) gate-level delay and path-level delay correlations.

2.1.2 Monte Carlo methods

The ***Monte Carlo*** (MC) technique is the other important approach for SSTA. Given a model of process variations, the classical MC-based method draws random samples in the process parameter space, and addresses the timing verification problem with circuit simulation tools. The main hurdle is the high computational cost. Thus, MC methods have been mostly relegated to a supporting role as the "gold standard" for validating the accuracy of proposed parametric SSTA methods.

However, MC techniques have recently attracted new attention as a candidate for a reliable and accurate timing verification, because MC techniques can account for any complicated model if one is willing to accept its excessive runtime costs. Moreover, the task of developing and integrating MC techniques is easy, because the available CTA engines can mostly be reused in developing new MC-based SSTA tools.

In recent works [25] – [27], the authors use techniques, such as ***importance sampling, Latin hypercube sampling***, to improve the performance of MC-based methods. However, more research is required to examine if these sampling techniques are effective in the domain of timing analysis.

2.2 BASIC STATISTICAL MODELS AND TECHNIQUES

The majority of SSTA methods proposed in the last few years are based on parametric models. Thus, in this section, we focus on these parametric models and related techniques. In general, a parametric timing method, either block-based or path-based, contains the following three basic steps:

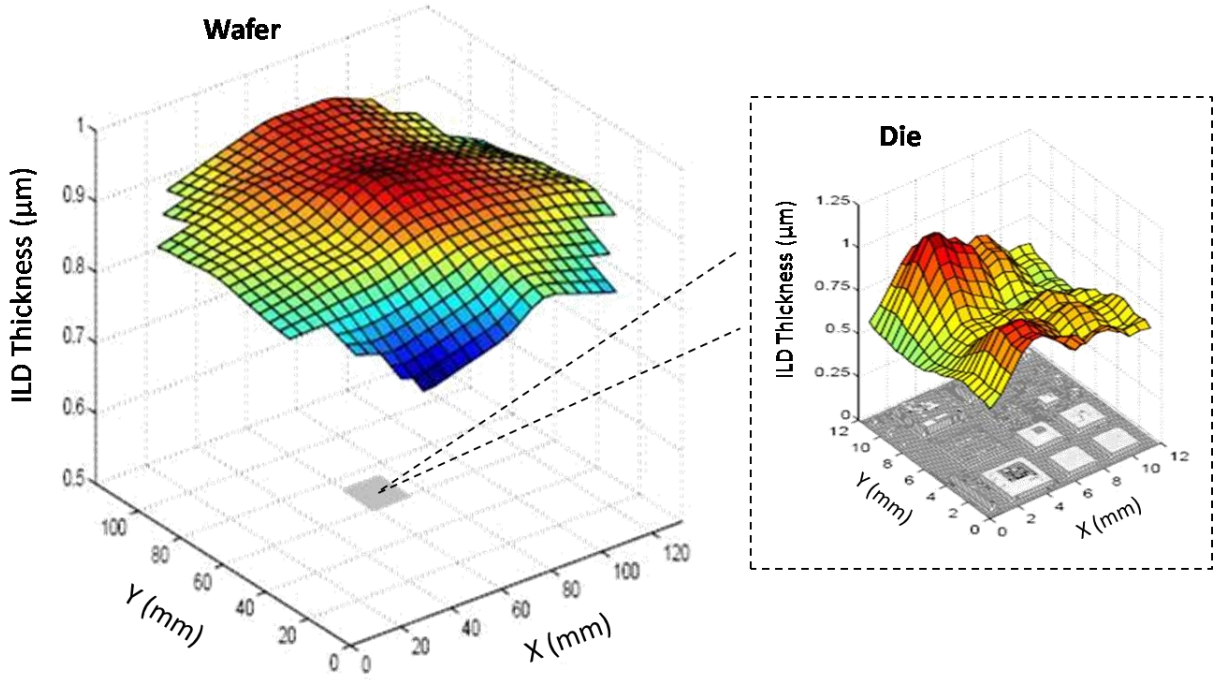
a) process variations modeling;
b) gate-level performance modeling;
c) propagation techniques.

2.2.1 *Process variations modeling*

For the purpose of design analysis, it is beneficial to divide the process variations into two categories: inter-die and intra-die variation. ***Inter-die variation*** is the variation that occurs from die-to-die and wafer-to-wafer. ***Intra-die variation*** is the component of variations that causes parameters to vary across different locations within a single die. For example, the inter-die and intra-die variation of inter-level dielectric thickness T_{ILD} are illustrated in FIGURE 2.3. It is reasonable to capture these two types of variations separately as:

$$T_{ILD} = T_{ILD,nom} + \Delta T_{ILD,inter} + \Delta T_{ILD,intra} \tag{2.1}$$

where $T_{ILD,nom}$ is the nominal value of ILD thickness, $\Delta T_{ILD,inter}$ is the variation due to inter-die sources, and $\Delta T_{ILD,intra}$ is the intra-die variation.



(a) Inter-die variation

(b) Intra-die variation

FIGURE 2.3 Variation in ILD thickness across the wafer and across the die [6]

The simplest way to model process variations is to consider the intra-die variation as a random variable $\Delta T_{ILD,intra}$ independent of the random variable $\Delta T_{ILD,inter}$, so that for any two gates k_1 and k_2 in the same die, we have:

$$\begin{cases} \Delta T_{ILD,inter\,,k_1} = \Delta T_{ILD,inter\,,k_2} \\ cor\left(\Delta T_{ILD,intra\,,k_1}, \Delta T_{ILD,intra\,,k_2}\right) = 0 \end{cases} \tag{2.2}$$

According to FIGURE 2.3(b), the variation across the die shows a spatial trend. So a better solution is to divide further the intra-die variation into two components: spatially correlated component and random component. Then EQUATION (2.1) can be rewritten as:

$$T_{ILD} = T_{ILD,nom} + \Delta T_{ILD,inter} + \Delta T_{ILD,spl} + \Delta T_{ILD,ran} \tag{2.3}$$

The spatial component $\Delta T_{ILD,spl}$ in EQUATION (2.3) is a function of the location on the die. Among the techniques to model spatial variation, the grid model [11] and the quad-tree model [12] are usually quoted in papers on SSTA.

For the ***grid model*** [11], the die region is partitioned into *N* squares, as shown in FIGURE 2.4, each of which is associated with one spatially correlated random variable. This implies that the spatial component is the same at any location on a given square. As gates close to each other are more likely to have similar characteristics than those placed far away, it is reasonable to assume high correlation among spatial components in close squares and low correlation in far-away squares. In FIGURE 2.4, according to the locations of gates k_1, k_2, k_3, k_4, we have:

$$\begin{cases} \Delta T_{ILD,spl,k_1} = \Delta T_{ILD,spl,k_2} \\ cor\left(\Delta T_{ILD,spl,k_1}, \Delta T_{ILD,spl,k_3}\right) \approx 1 \\ cor\left(\Delta T_{ILD,spl,k_1}, \Delta T_{ILD,spl,k_4}\right) \approx 0 \end{cases} \tag{2.4}$$

In addition, another assumption for the grid model is that spatial correlation exists only among the same type of parameters in different squares and there is no spatial correlation between different types of parameters. For example, T_{ILD} are independent with other parameters such as L_{eff} or T_{ox} in any square.

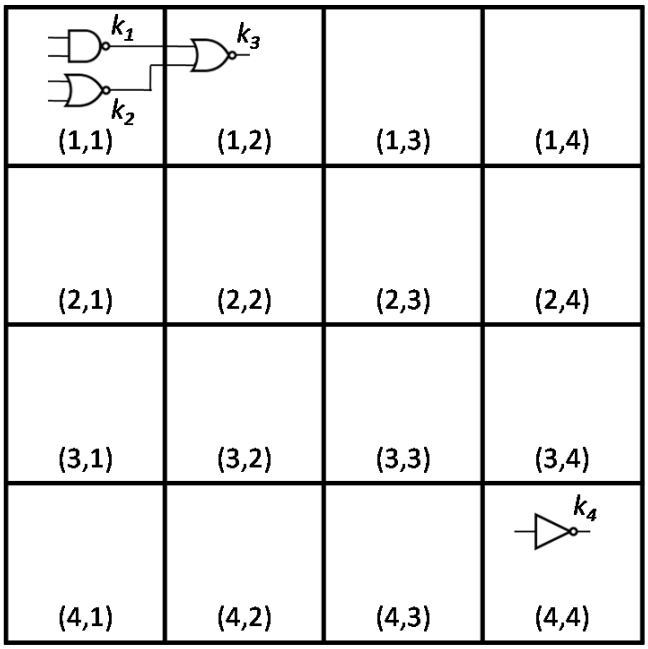


FIGURE 2.4 An example of the grid model

For the ***quad-tree model***, proposed in [12], the die area is divided into several regions using quad-tree partitioning, where at level i, the die is partitioned into $2^i \times 2^i$, $(i = 0, 1, 2, \dots)$ squares. All of the squares of the tree are associated with an independent random variable. A three-level tree is illustrated in FIGURE 2.5.

For the process parameter T_{ILD}, an independent random variable $\Delta T_{ILD,i,j}$ is associated with the variation in square j at level i. For example, in FIGURE 2.5, the spatial variation in T_{ILD} of gate k_1, k_2 is express as follows:

$$\begin{cases} \Delta T_{ILD,spl,k_1} = \Delta T_{ILD,0,1} + \Delta T_{ILD,1,1} + \Delta T_{ILD,2,1} \\ \Delta T_{ILD,spl,k_2} = \Delta T_{ILD,0,1} + \Delta T_{ILD,1,4} + \Delta T_{ILD,2,11} \end{cases} \tag{2.5}$$

In EQUATION (2.5), the occurrence of the same random variable $\Delta T_{ILD,0,1}$ in both formulas models the spatial correlation between $\Delta T_{ILD,spl,k_1}$ and $\Delta T_{ILD,spl,k_2}$.

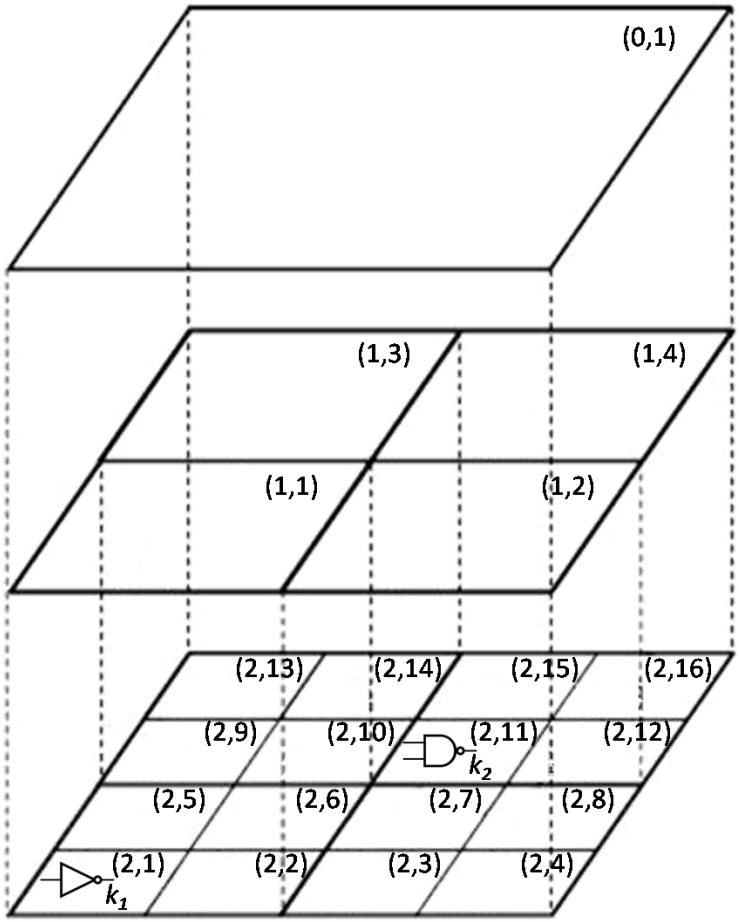


FIGURE 2.5 An example of the quad-tree model

2.2.2 Gate-level performance modeling

Unlike CTA that approximates the function $f_{type\ ,pin\ ,edge}$ in EQUATION (1.1) with lookup tables and the bilinear interpolation technique, parametric SSTA models gate delay with polynomials derived from Taylor expansion. Most of the parametric models make the assumptions that:

a) V, T and τ_{in} are at their corresponding corners;

b) C_{out} is a constant;

c) probability distribution F_l of $p_l, (l = 1, 2, \dots, L)$ is known.

Then, the function $f_{type,pin,edge}$ can be approximated using the first or second order Taylor expansion:

$$gd \approx gd_{nom} + \sum_{l=1}^{L} a_l \cdot \Delta p_l \tag{2.6}$$

$$gd \approx gd_{nom} + \sum_{l=1}^{L} a_l \cdot \Delta p_l + \sum_{l=1}^{L} b_l \cdot \Delta p_l^2 + \sum_{\forall l_1 \neq l_2}^{L} c_{l_1 l_2} \cdot \Delta p_{l_1} \Delta p_{l_2} \tag{2.7}$$

where gd_{nom} is the nominal value of d; a_l and b_l are the first and the second order sensitivities of gd to Δp_l, respectively; and $c_{l_1 l_2}$ are the sensitivity to the joint variation of Δp_{l_1} and Δp_{l_2}. When all Δp_l are assumed to be Gaussian random variables, EQUATION (2.6) is called the canonical model, and has been widely used for SSTA [11] – [13]; whereas EQUATION (2.7) is called the quadratic model, and has been studied in [14] – [16], [19] – [20]. However, these parametric models based on Gaussian assumptions are limited in their modeling capability because not all process variations follow the Gaussian distribution. Therefore, [17] – [18] extend the work by adding non-Gaussian terms to EQUATION (2.6).

2.2.3 Propagation techniques

After the gate-level performances of all circuit components have been modeled, circuit delay needs to be determined. Essential operations are the SUM and the MAX of random variables. The gate-to-gate delay correlation, which is difficult to estimate, needs to be considered for these operations. In addition, the statistical MAX operation is computationally expensive to be determined exactly, which is one of the most challenging problems in the domain of SSTA.

In the SUM operation, if both X and Y are random variables, then $Z = X + Y$ will also be a random variable whose mean and variance can be found as:

$$\begin{cases} \mu_Z = \mu_X + \mu_Y \\ \sigma_Z^2 = \sigma_X^2 + \sigma_Y^2 + \rho_{XY} \cdot \sigma_X \sigma_Y \end{cases} \tag{2.8}$$

where ρ_{XY} is the correlation between X and Y.

As the MAX operation is nonlinear, $W = max(X, Y)$ is not a Gaussian random variable even when both X and Y are Gaussians and independent. In [30], the author proposes a moment matching approach to approximate the distribution of W with that of a Gaussian random variable $\widehat{W}$. Define $V = X - Y$ and the following standard Gaussian ***Probability Density Function*** (PDF) and ***Cumulative Distribution Function*** (CDF):

$$\begin{cases} \varphi(x) = \dfrac{1}{\sqrt{2\pi}} \cdot e^{\frac{-x^2}{2}} \\ \\ \Phi(x) = \displaystyle\int_{-\infty}^{x} \varphi(u)du \end{cases} \tag{2.9}$$

Then, $\widehat{W}$ is given by:

$$\widehat{W} = \Phi\left(\frac{\mu_V}{\sigma_V}\right) \cdot X + \left(1 - \Phi\left(\frac{\mu_V}{\sigma_V}\right)\right) \cdot Y + \varphi\left(\frac{\mu_V}{\sigma_V}\right) \cdot \sigma_V \tag{2.10}$$

where

$$\begin{cases} \mu_V = \mu_X - \mu_Y \\ \sigma_V = (\sigma_X^2 + \sigma_Y^2 - \rho_{XY} \cdot \sigma_X \sigma_Y)^{1/2} \end{cases} \tag{2.11}$$

2.3 CHALLENGES FOR SSTA

Although SSTA has made significant progresses in the past six years, it is still in the neonatal state and much work needs to be done to improve it. To date, most SSTA researchers have mainly focused on the basic SSTA techniques – the SUM and MAX operations required for the propagation of arrival times from the source node to the sink node of the timing graph. For wider adoption of SSTA, its capabilities must be extended to match the current state of CTA, such as a corresponding SSTA design flow including statistics-based optimization. For this reason, this section presents not only the weaknesses to overcome, but also the outlook of SSTA.

2.3.1 Weaknesses of existing models and techniques

There are many sources of weaknesses in the existing parametric SSTA techniques and most of them derive from the model used for the analysis. Some of the common sources can be classified into the following categories:

- ***Unsatisfying models of process variations***

  Most of the initial work in SSTA assumed Gaussian distributions for process parameters. Actually, some of them follow significantly non-Gaussian distributions. For example, via resistances exhibit an asymmetric probability distribution [17], and the dopant concentration density seems to be well modeled by a Poisson distribution [18]. Thus, under the Gaussian assumption for all process parameters, the accuracy of timing analysis is not guaranteed.

  Another modeling problem is the correlation between process parameters. The models presented in SECTION 2.2.1 are only suitable to capture variations of the same type of parameter.

  Besides, the availability of data to construct statistical process models remains scarce.

- ***Limitations of gate delay models***

  The majority of the existing parametric SSTA techniques are based on polynomial model of timing performance. Many of these techniques consider only few process parameters, like V_{th}, L_{eff}, t_{ox}, and have reported high modeling accuracy. However, due to the increase in process variability, parametric models with more parameters are expected to be necessary to achieve the same accuracy [31].

  In addition, the intrinsic nature of timing performance depending on process parameters is complex and nonlinear. Consequently, linear models are not enough for acceptable approximations. As for second order models, the cost of better accuracy is the much higher computational complexity. As an example, a quadratic expression with 30 uncorrelated

variables has over 400 terms if cross-terms are considered. Therefore, existing parametric SSTA methods need to be revisited.

Apart from the tradeoff between accuracy and runtime, another common limitation of existing parametric models, according to EQUATION (1.1), is that some effects, besides those listed in SECTION 2.1.1, are ignored or simplified; to name a few:

a) the random variations in input slope and output load,
b) the time-dependent variations in supply voltage and temperature,
c) the effects of input pin on gate delay,
d) interdependency among input/output edges and gate delay.

- ***Inaccurate approximation of MAX operation***

The linear approximation of MAX operation in EQUATIONS (2.10) – (2.11) is simple and independent of parametric models, but its accuracy is not satisfying. Even if arrival times are assumed to be Gaussian distributed, the MAX of them will be a non-Gaussian distribution. The error of this approximation will be larger if the input arrival times have similar means and dissimilar variances [30]. This case occurs when two converging paths with similar nominal values have a different number of gates. A simple example is illustrated in FIGURE 2.6. Suppose two independent Gaussian random variables X and Y have the same zero mean and different variances, i.e. $X \sim N(0, \sigma_X^2)$, $Y \sim N(0, \sigma_Y^2)$ with $\sigma_X^2 \neq \sigma_Y^2$. The density of W and $\widehat{W}$ are respectively, from Monte Carlo simulation and the approximation in EQUATIONS (2.10) – (2.11), shown in FIGURE 2.6. The error of the estimator $\widehat{W}$ is significant.

In addition, with non-Gaussian variations to gate-level performances found by the authors of [14] – [18], the linear approximation is even worse. The new corresponding MAX approximations in these papers are closely related to their proposed parametric models. Hence, a model-independent MAX approximation that can operate on non-Gaussian random variables is required.

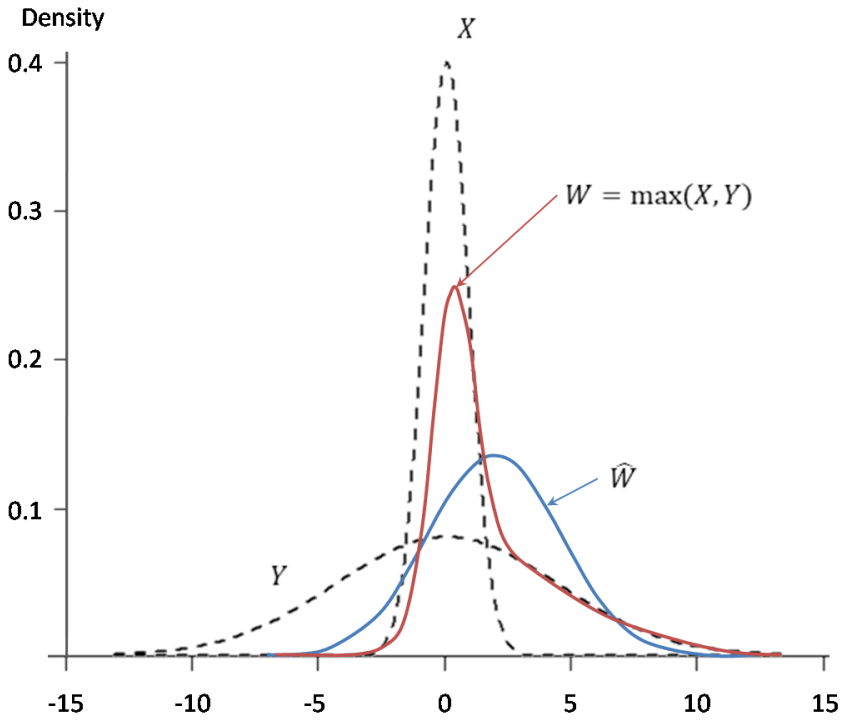


FIGURE 2.6 Accuracy of the linear approximation of the MAX operation

In [32], timing performances are modeled with skew-normal distributions which embed the Gaussian distribution to allow for non-zero skewness. Empirically, this technique offers better accuracy than the linear approximation of MAX for a broad set of models. However, the computational cost of such an approximation is high.

2.3.2 Outlook for SSTA

CTA has evolved over the last two decades and is able to handle a number of practical issues, like crosstalk noise, power and ground noise, clock skew, etc. However, most SSTA researchers have, to date, mainly focused on the basic statistical timing techniques: process models, gate-level performance models, and approximations of MAX. For wider adoption of SSTA, these techniques must be perfected to match the mature state of CTA.

Recently, a few methods have been proposed to address some of these issues in SSTA. The authors in [33] propose a statistical gate-delay modeling technique that considers multiple input switching. In [34], a probabilistic collocation-based method is presented to efficiently construct

statistical gate-delay models. Finally, a statistical framework for modeling the effect of crosstalk-induced coupling noise on timing was presented in [35].

In addition to crosstalk noise, SSTA of sequential circuits is another area that still requires significant investigation. Several issues related to sequential timing, such as accurate modeling of variations and dependences in the clock tree, clock skew analysis and clock schedule verification for multiple clock domains, still need to be resolved. Recently, several research efforts have focused on these issues [36] – [37].

Finally, for statistics-based optimization, efficient methods for slack computation are needed. Some initial methods for slack computation in SSTA are given in [38] – [39]. Other topics, like gate sizing and buffer insertion, are addressed in [40] – [42].

To summarize, SSTA must move beyond pure timing analysis to yield analysis and optimization of circuit design to be truly useful for the designers. If the data from industry shows that SSTA-based designs have substantially higher manufacturing yield than CTA-based designs, the wide adoption of SSTA will be guaranteed.

2.4 SUMMARY

MC-based SSTA methods are accurate, whereas parametric methods are of a very low computational cost. In general, a parametric method uses the first or second order Taylor approximation to model gate delay based on a process variation model, like grid model and quad-tree model. Then, circuit delays are computed by approximating the MAX operation with a linear function. These models of process variations and gate delay, plus MAX approximations still have many weaknesses to overcome. SSTA is promising in nature, but a lot of work needs to be done for its wide adoption.

CHAPTER

3

PATH-BASED SSTA FRAMEWORK

This chapter first describes the flow of the proposed SSTA framework. After the introduction of conditional moments in SECTION 3.2, we focus on moments propagation in SECTION 3.3, which is the key part of our SSTA engine. SECTION 3.4 shows how to compute path delay distributions. SECTION 3.5 discusses the estimation of delay correlations. Finally, the validation and a discussion of this SSTA framework are given.

Monte Carlo (MC) methods are accurate, but suffer from their very high computational cost. On the contrary, although the existing parametric methods of SSTA are efficient, industry and researchers are doubtful of their accuracy because of diverse weaknesses and limitations, as presented in CHAPTER 2. A good compromise would be a method that can make an acceptable trade-off between accuracy and efficiency. In this chapter, we present our path-based SSTA framework, which offers high efficiency while somewhat keeping the advantage of MC methods. Such features are achieved by propagating iteratively means and variances of cell[1] delay with the help of conditional moments. These moments, conditioned on input slope and output load, are pre-characterized by MC simulations, and organized as a tree of lookup tables, called a ***statistical timing library***. This characterization step is a one-time job, i.e. the high time-cost simulation is only needed to build the statistical timing library. This creates a semi-MC framework that allows us to:

a) avoid cell delay modeling errors;
b) take into account the effects on cell delay: input pin, output edge, input slope, and output load;
c) deal with a large number of process parameters having any type of distribution.

3.1 FLOW OF THE PATH-BASED SSTA FRAMEWORK

For ease of description, the SSTA flow is divided into four parts, as shown in FIGURE 3.1:

- ***Setup*** – construct a statistical timing library;
- ***Input*** – define environmental conditions and extract a set of candidate paths for a given circuit design;
- ***SSTA engine*** – compute the circuit delay;
- ***Output*** – generate the statistical timing report.

Details about this flow are given in the rest of this section.

[1] A cell is either a gate or a flip-flop.

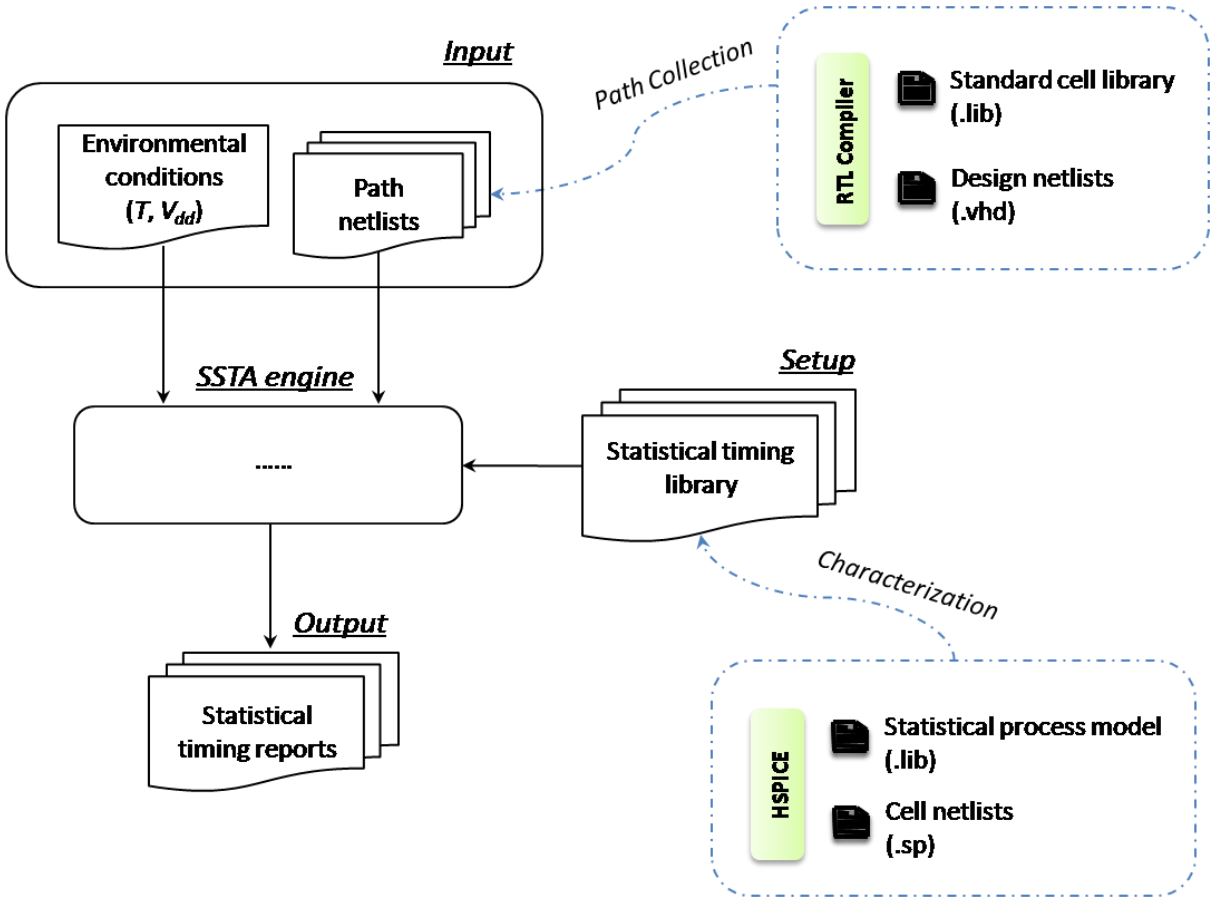


FIGURE 3.1 Flow of our path-based SSTA framework

3.1.1 Setup

This initial step of the flow is to prepare a statistical timing library that feeds the SSTA engine. FIGURE 3.1 indicates that the characterization of the library is done with a statistical process model and the cell netlists under HPSICE [43], which provides the necessary data to construct the library.

A ***cell netlist*** define the structure and the default characteristics of the cell. A ***statistical process model*** describes process parameters with probability distributions, like Gaussian, Uniform or Poisson. The parameters of these distributions are estimated by empirical data from existing IC.

In this thesis, the 130 nm and 65 nm statistical process models provided by ST Microelectronics[2], are described as follows:

a) For each process parameter p_l, as in SECTION 2.2.1, we have:

$$p_l = p_{nom,l} + \Delta p_{inter,l} + \Delta p_{intra,l} \tag{3.1}$$

where $l = 1, 2, \ldots L$; $p_{nom,l}$ is the nominal value of p_l; the intra-die random variable $\Delta p_{intra,l}$ is independent of the inter-die random variable $\Delta p_{inter,l}$. Note that most of the process parameters only have the inter-die component because the intra-die variation is small enough to be neglected.

b) The probability distributions of $\Delta p_{inter,l}$ and $\Delta p_{intra,l}$ are known.

c) For any $l_1 \neq l_2, (l_1, l_2 = 1, 2, \ldots L)$, p_{l_1} and p_{l_2} are independent.

d) For any two cells k_1 and k_2 in the same die, $\Delta p_{intra,l,k_1}$ and $\Delta p_{intra,l,k_2}$ are independent, i.e. there is no spatial correlation.

TABLE 3.1 gives the information about the cell netlists and the statistical process models. In the 130 nm technology, all intra-die variations $\Delta p_{intra,l}, (l = 1, 2, \ldots L)$ are neglected.

TABLE 3.1 Information about the cell netlists and the statistical process models

technology	cell netlists	BSIM model	number of statistical process parameters L	
			inter-die	intra-die
130 nm	CORE9GPLL	v3	52	0
65 nm	CORE65LPHVT	v4	76	2

Knowing the distribution of each p_l, we do MC simulation under various conditions and organize the output data as a tree of lookup tables. In FIGURE 3.2, the statistical timing library has a tree structure with levels: cell type, ***input/output*** (I/O) pin, I/O edge, temperature, supply voltage and timing variables. The tree leaves are lookup tables, each of which contains an input slope index, an output load index and moments conditioned on these indices.

[2] an Italian-French electronics and semiconductor manufacturer

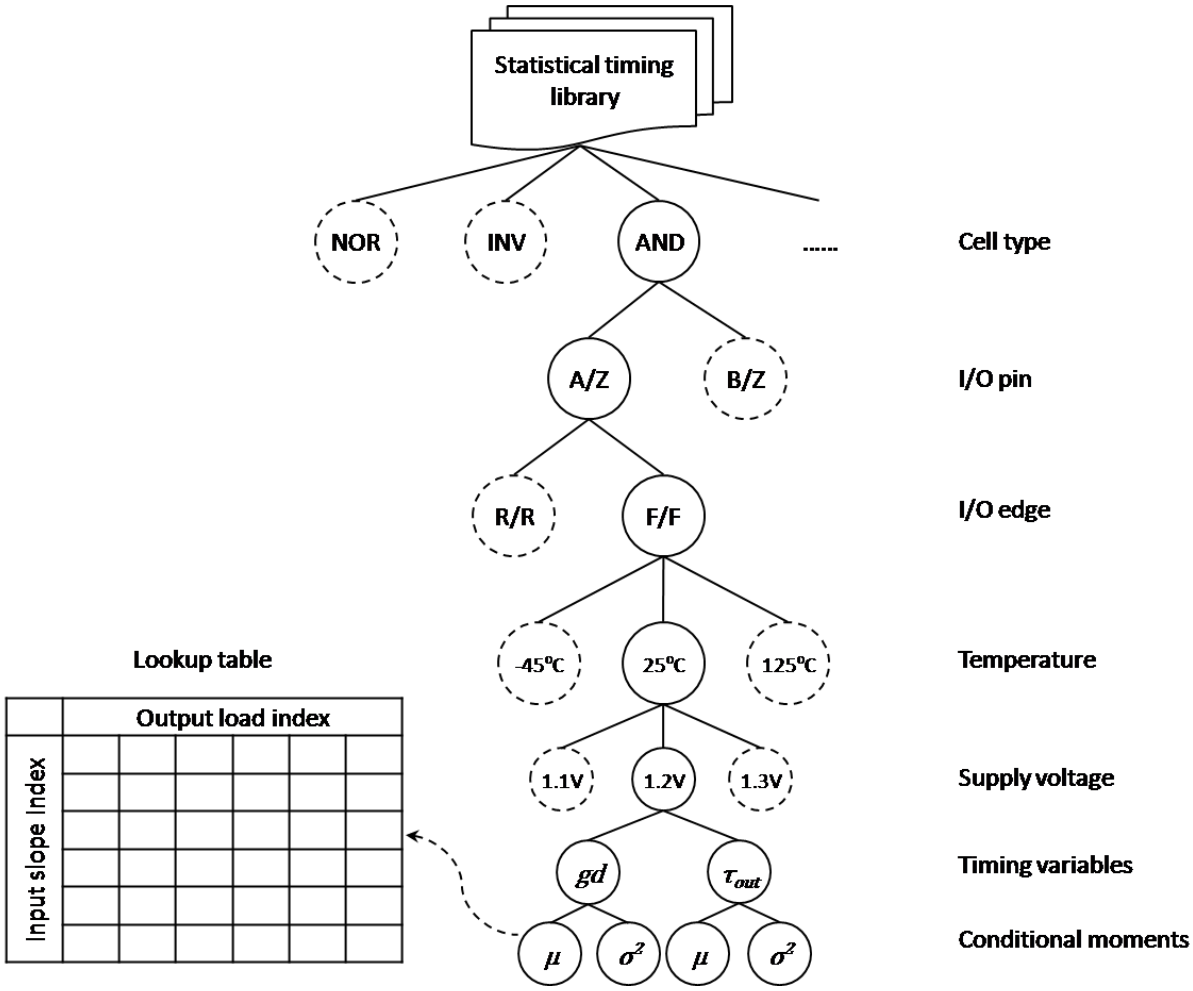


FIGURE 3.2 Structure of the statistical timing library

In FIGURE 3.3, a lookup table and the corresponding function that it approximates are given. The input slope index $\tau_{in} = \tau_1, \tau_2, \dots, \tau_9$ and the output load index $C_{out} = c_1, c_2, \dots, c_6$ are chosen according to:

- the upper and lower limits of τ_{in} and C_{out},
- the sensitivities of conditional moments on τ_{in} and C_{out}.

For any couple (τ_m, c_n), $(m = 1, 2, \dots, 9)$ and $(n = 1, 2, \dots, 6)$, the output slope mean μ_{mn} conditioned on τ_m and c_n is estimated with data from simulations. Then for any point in the rectangular region $[\tau_1, \tau_9] \times [c_1, c_6]$, its conditional output slope mean is obtained using bilinear interpolation. The corresponding conditional variance is computed in a similar way.

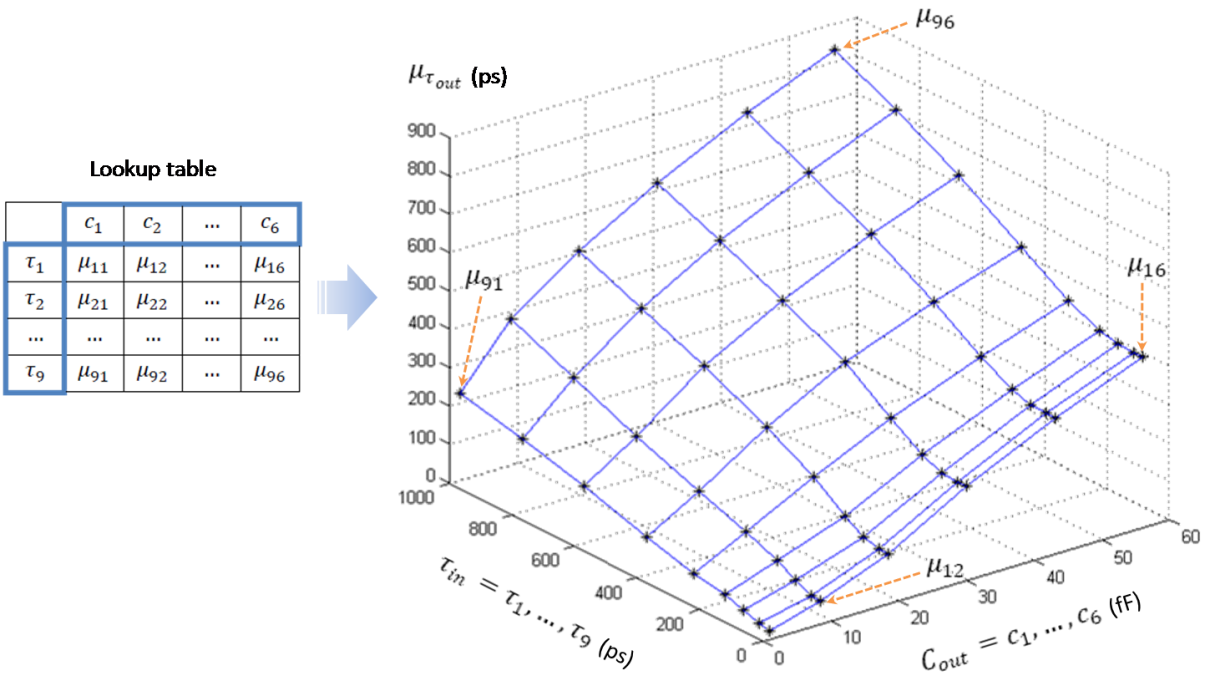


FIGURE 3.3 Illustration of approximating a complicated function with a lookup table

As discussed above, the library has taken into account all factors that affect cell delay, because:

a) Process variations are captured during simulation, and contained in conditional variances;
b) Cell type, input pin, output edge, temperature and supply voltage are tree levels;
c) Input slope and output load are indices of lookup tables.

3.1.2 Input

The input for the SSTA engine includes environmental conditions and path netlists. Environmental conditions are temperature and supply voltage. As mentioned in SECTION 1.1.3, the task of modeling time-dependent environmental variations is difficult. Thus, the statistical timing library only supports temperatures $-45℃$, $25℃$, $125℃$ and supply voltages $1.1V$, $1.2V$, $1.3V$.

Path netlists is the set of critical paths. In this thesis, given a circuit design, we first implement a CTA, and then collect the top N paths in decreasing order of path delay [44]. This work of path collection is done using RTL Compiler [45]. Obviously, the accuracy of the SSTA engine will improve if the number of paths N increases. However, considering computational cost, we need

to determine N carefully. Besides, even though N has been well chosen, different subsets of all possible paths could lead to significantly different results. Consequently, the efficient generation of a set of candidate paths in a circuit is central to path-based methods.

3.1.3 SSTA engine

FIGURE 3.4 shows the procedure of the SSTA engine. Given a set of N paths, the engine computes the path delay distributions one by one. Then, the circuit delay cd is computed by:

$$cd = max(pd_1, pd_2, \dots, pd_N) \tag{3.2}$$

Assuming that path delays are Gaussian distributed, the distribution of cd is computed using the algorithms in [46], which is based on the linear approximation of MAX in EQUATIONS (2.10) – (2.11). We know that path delay is obtained by summing all delays of cells on a path. Hence, even if cell delays are not Gaussians, it is still reasonable to set Gaussian distributions to path delays as a first approximation, because a sum of independent random variables rapidly converges (for most practical correlation structures involved in circuit delay computation) to a Gaussian random variable due to the ***central limit theorem*** [47].

As for cell-level delays, we make no assumption on their distributions, and just propagate means and variances. For cell k, cell type, I/O pin, I/O edge, temperature T, supply voltage V_{dd} and output load $C_{out,k}$ are known from the procedure of path collection; input slope of a cell is the output slope of the previous cell, i.e. $\mu_{\tau_{in,k}} = \mu_{\tau_{out,k-1}}$ and $\sigma^2_{\tau_{in,k}} = \sigma^2_{\tau_{out,k-1}}$. Then, the moments $\mu_{gd_k}, \sigma^2_{gd_k}, \mu_{\tau_{out,k}}, \sigma^2_{\tau_{out,k}}$ are computed with the help of lookup tables and bilinear interpolation.

3.1.4 Output

A statistical timing report includes the information as follows:

a) cell-level results: cell delay means and variances, cell-to-cell delay correlation;
b) path-level results: path delay distributions, path-to-path delay correlation;
c) circuit-level results: circuit delay distribution.

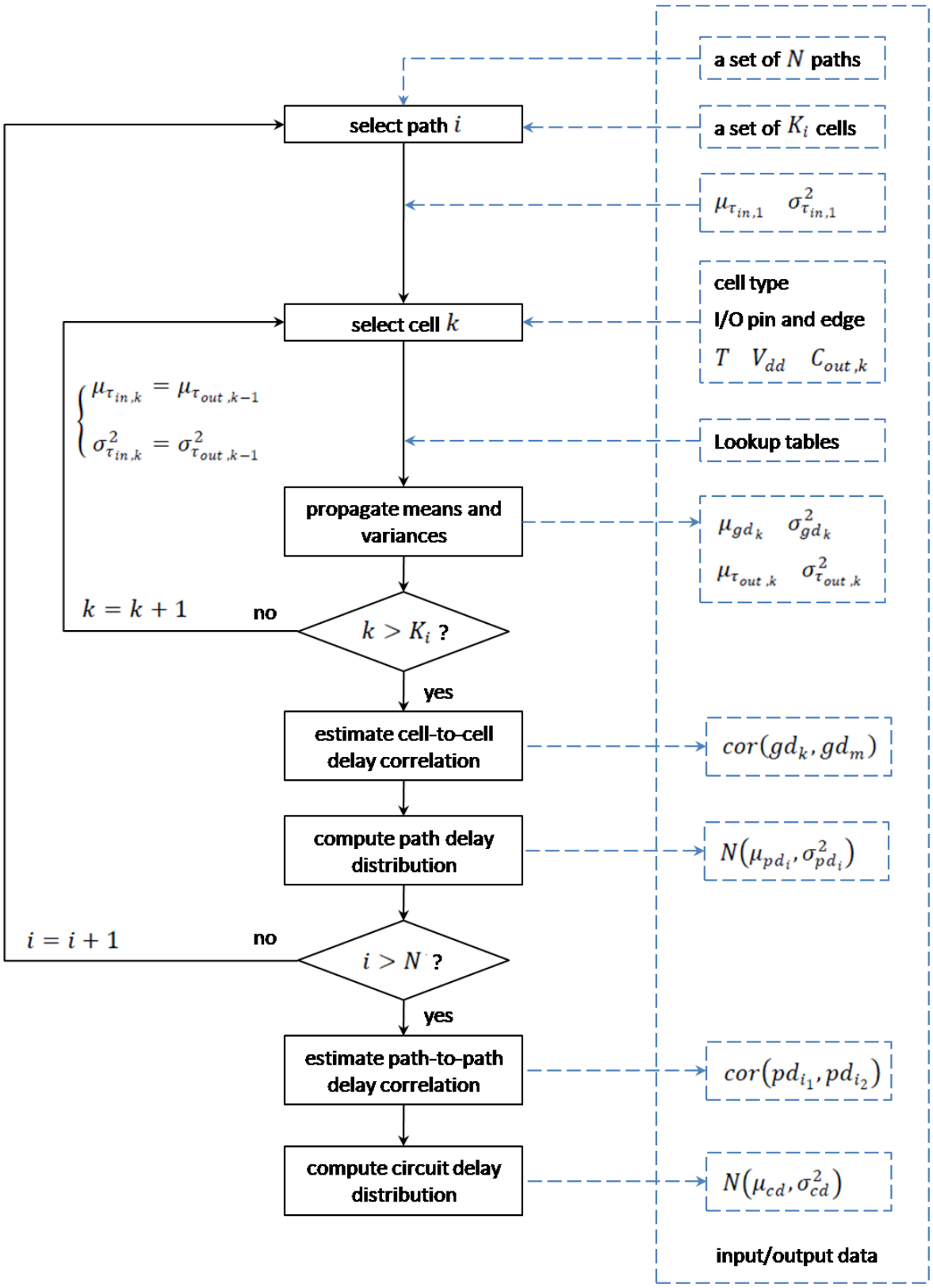


FIGURE 3.4 Procedure of the SSTA engine

3.2 CONDITIONAL MOMENTS

The mean and variance of a random variable X, if they exist, are respectively denoted as $E(X)$ and $Var(X)$, where $Var(X) = E(X^2) - E^2(X)$. They are also called ***moments*** of X.

A ***conditional moment*** is the moment of one random variable conditioned on the value of another random variable. If X and Y are two random variables, then the ***conditional mean*** $E(X|Y = y)$ is the mean of X given the value $Y = y$. In our case, X is continuous while Y can be either discrete or continuous. Given the ***Probability Density Function*** (PDF) of X conditioned on $Y = y$, denoted as $f(x|y)$, we define:

$$E(X|Y = y) = \int_{-\infty}^{\infty} x \cdot f(x|y)dx \tag{3.3}$$

Unlike the conventional mean $E(X)$, which is a constant for a specific probability distribution, $E(X|Y = y)$ is a function of y, that is to say, the conditional mean varies along with the value taken by Y.

Similarly, the ***conditional variance*** $Var(X|Y = y)$ is the variance of X given the value $Y = y$, defined by:

$$Var(X|Y = y) = E(X^2|Y = y) - E^2(X|Y = y) \tag{3.4}$$

With these definitions of conditional moments, the mean and the variance of X can be decomposed as:

$$\begin{cases} \mu_X = E(X) \ = E[E(X|Y = y)] \\ \sigma_X^2 = Var(X) = E[Var(X|Y = y)] + Var[E(X|Y = y)] \end{cases} \tag{3.5}$$

The proofs of these two decompositions are given in [48]. Next, from EQUATION (3.5), we derive two groups of equations adapted to the cases where Y follows respectively a discrete and continuous distribution.

If Y follows a discrete probability distribution:

$$\begin{cases} Pr(Y = y_i) = \alpha_i > 0 & i = 1, \dots, I \\ \sum_{i=1}^{I} \alpha_i = 1 \end{cases} \tag{3.6}$$

then we have:

$$\begin{cases} \mu_X = \sum_{i=1}^{I} \alpha_i \cdot E(X|Y = y_i) \\ \sigma_X^2 = \sum_{i=1}^{I} \alpha_i \cdot \{Var(X|Y = y_i) + [E(X|Y = y_i) - E(X)]^2\} \end{cases} \tag{3.7}$$

Here is a concrete illustration of these two decompositions. Suppose that X follows a continuous distribution with PDF $f(x)$ and Y follows the discrete distribution in EQUATION (3.6). In addition, suppose there exists some dependency between X and Y. We draw a sample of (X, Y) from their joint distribution and divide it into I groups, each of which has the same value $y_i, (i = 1, \dots, I)$. In this case, $E(X|Y = y_i)$ and $Var(X|Y = y_i)$ represent respectively the mean and variance of X in group y_i. Then, $E(X)$ is the sum of all $E(X|Y = y_i)$ weighted by α_i. As for $Var(X)$, it consists of two parts: variance between groups $[E(X|Y = y_i) - E(X)]^2$ and variance within group $Var(X|Y = y_i)$. In other words, total variance can be explained by the sum of inter-variance and intra-variance both weighted by α_i.

On the other hand, if Y follows a continuous distribution with PDF $f(y)$, then we have:

$$\begin{cases} \mu_X = \int E(X|Y = y) \cdot f(y)dy \\ \sigma_X^2 = \int \{Var(X|Y = y) + [E(X|Y = y) - \mu_X]^2\} \cdot f(y)dy \end{cases} \tag{3.8}$$

EQUATIONS (3.7) – (3.8) give an alternative to compute the mean and variance of X if these two moments cannot be obtained directly with traditional methods. These equations require some dependency between X and Y, which allows implementing the idea of moments propagation.

3.3 MOMENTS PROPAGATION

This section presents the technique to propagate moments of timing variables iteratively along a timing path. We assume that all timing variables follow continuous distributions.

Let us define the problem of moments propagation. Suppose the context is known, including: cell type, I/O pin, I/O edge, supply voltage, temperature, and output load. Then, for the considered cell, given the moments $\mu_{\tau_{in}}$, $\sigma^2_{\tau_{in}}$ of input slope, we seek to get the output slope moments $\mu_{\tau_{out}}$, $\sigma^2_{\tau_{out}}$ and the cell delay moments μ_{gd}, σ^2_{gd}.

FIGURE 3.5 illustrates the procedure of propagation. Knowing $\mu_{\tau_{in,1}}$ and $\sigma^2_{\tau_{in,1}}$, we look up the statistical timing library according to the context, and do bilinear interpolations for moments of output slope and cell delay conditioned on input slope and output load, i.e. $E(\tau_{out}|\tau_{in,1}, C_{out,1})$, $Var(\tau_{out}|\tau_{in,1}, C_{out,1})$, $E(gd_1|\tau_{in,1}, C_{out,1})$, and $Var(gd_1|\tau_{in,1}, C_{out,1})$. After that, $\mu_{\tau_{out,1}}$, $\sigma^2_{\tau_{out,1}}$, μ_{gd_1} and $\sigma^2_{gd_1}$ are computed by equations presented later. Lookup, interpolate and compute, these three steps are repeated for the second cell by taking $\mu_{\tau_{in,2}} = \mu_{\tau_{out,1}}$ and $\sigma^2_{\tau_{in,2}} = \sigma^2_{\tau_{out,1}}$.

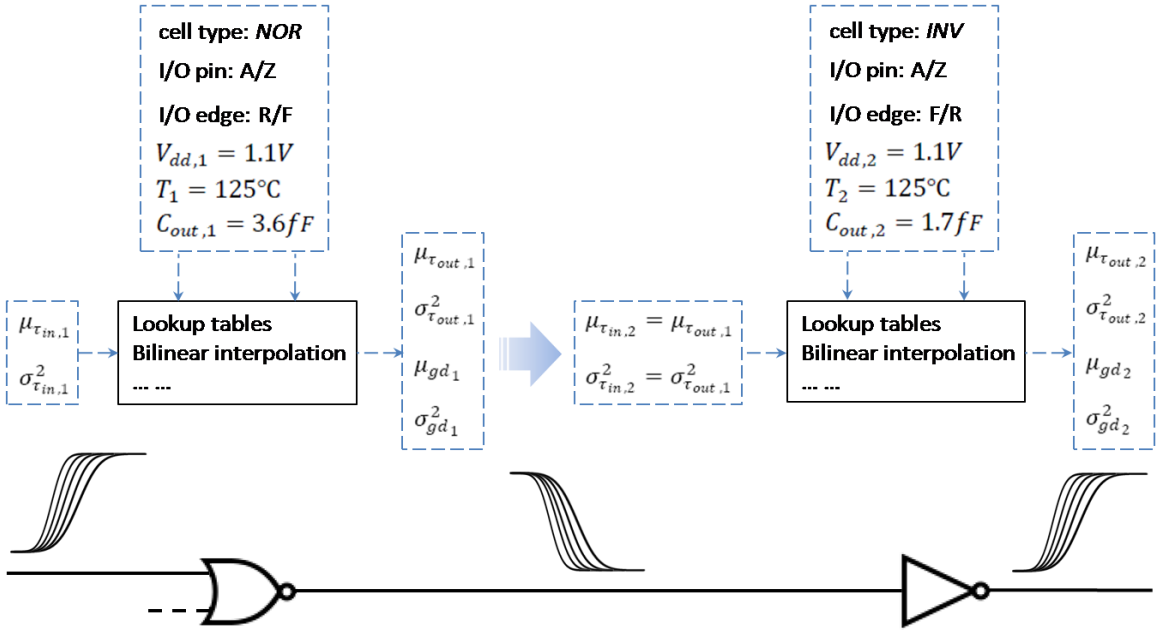


FIGURE 3.5 Illustration of moments propagation

Note that only the moments of timing variables instead of distributions are known. For example, cell delay may follow any continuous distribution defined with two parameters on condition that its mean and variance exist, like Uniform, Gaussian, etc. In addition, the output load of any cell takes its nominal value, because its variation has been captured during timing characterization.

3.3.1 Interpolation

In FIGURE 3.3, the lookup table only provides conditional means μ_{mn} of τ_{out} for some finite number of τ_{in} and C_{out}, where $\mu_{mn} = E(\tau_{out}|\tau_{in} = \tau_m, C_{out} = c_n)$. However, the function to approximate is continuous. In this case, a simple solution is the ***bilinear interpolation*** technique, which is an extension of linear interpolation for interpolating functions of two variables on a regular grid. The idea is to perform linear interpolation first in one direction, and then again in the other direction.

FIGURE 3.6 gives an example. For simplicity, we denote $E(\tau_{out}|\tau_{in} = \tau_m, C_{out} = c_n)$ as $E(\tau_{out}|\tau_m, c_n)$. Suppose $\tau_{in} = \tau \in [\tau_6, \tau_7]$ and $C_{out} = c \in [c_2, c_3]$, we first interpolate in the direction of C_{out}, and then in the direction of τ_{in}:

$$\begin{cases} E(\tau_{out}|\tau_6, c) \approx \dfrac{c_3 - c}{c_3 - c_2} \cdot E(\tau_{out}|\tau_6, c_2) + \dfrac{c - c_2}{c_3 - c_2} \cdot E(\tau_{out}|\tau_6, c_3) \\ E(\tau_{out}|\tau_7, c) \approx \dfrac{c_3 - c}{c_3 - c_2} \cdot E(\tau_{out}|\tau_7, c_2) + \dfrac{c - c_2}{c_3 - c_2} \cdot E(\tau_{out}|\tau_7, c_3) \end{cases} \tag{3.9}$$

$$E(\tau_{out}|\tau, c) \approx \frac{\tau_7 - \tau}{\tau_7 - \tau_6} \cdot E(\tau_{out}|\tau_6, c) + \frac{\tau - \tau_6}{\tau_7 - \tau_6} \cdot E(\tau_{out}|\tau_7, c) \tag{3.10}$$

With the lookup tables stored in statistical timing library, for any $\tau \in [\tau_1, \tau_9]$ and $c \in [c_1, c_6]$, we may get the following four conditional moments by bilinear interpolation: $E(\tau_{out}|\tau, c)$, $Var(\tau_{out}|\tau, c)$, $E(gd|\tau, c)$ and $Var(gd|\tau, c)$.

As mentioned above, output load c of any cell is set to its nominal value whereas input slope τ_{in} is a random variable. Thus, in SECTIONS 3.3.2 and 3.3.3, we only talk about the technique to capture variations of τ_{in}.

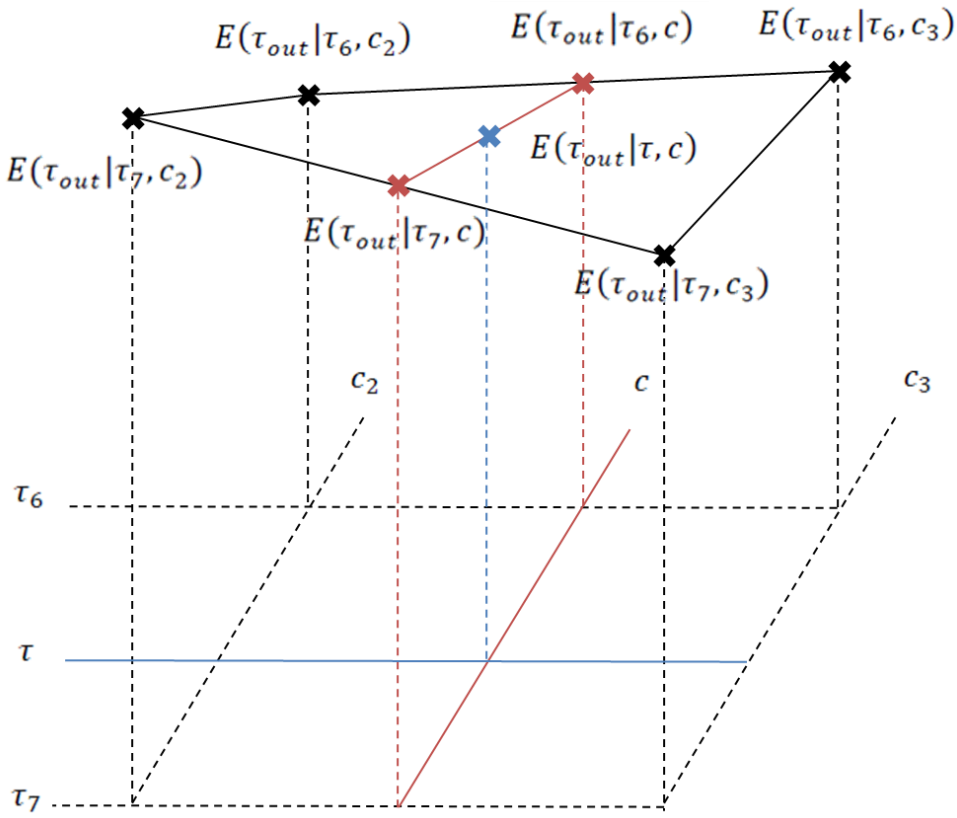


FIGURE 3.6 Illustration of bilinear interpolations

3.3.2 Discrete version

If X and Y in EQUATION (3.7) represent respectively the output slope τ_{out} and the input slope τ_{in} of a cell, then to compute $\mu_{\tau_{out}}$, $\sigma^2_{\tau_{out}}$, a discrete distribution of τ_{in} as in EQUATION (3.6) is necessary. However, at the beginning of SECTION 3.3, all timing variables were assumed to follow continuous distributions. Thus, to make use of EQUATION (3.7), we need to discretize the distribution of input slope.

For the purpose of discretization, the type of probability distribution must be known to compute the probability of each discrete point. Typically, we assume that all slopes are Gaussian distributed, which is a common assumption in most of the initial works on SSTA [11] – [13], [22]. Note that this Gaussian assumption is not set to cell delays, which are not required to be discrete.

To discretize $N(\mu_{\tau_{in}},\ \sigma_{\tau_{in}}^2)$, we divide the interval $\left[\mu_{\tau_{in}} - 3\sigma_{\tau_{in}},\ \mu_{\tau_{in}} + 3\sigma_{\tau_{in}}\right)$ into I equidistant parts: $[s_0,\ s_1),\ [s_2,\ s_3), \dots,\ [s_{I-1},\ s_I)$, where I is an even integer, $s_0 = \mu_{\tau_{in}} - 3\sigma_{\tau_{in}}$ and $s_I = \mu_{\tau_{in}} + 3\sigma_{\tau_{in}}$. Then the discrete distribution is determined by:

$$y_i = \frac{s_{i-1} + s_i}{2} \qquad i = 1, \dots, I \tag{3.11}$$

$$\alpha_i = \begin{cases} \int_{-\infty}^{s_1} f(\tau_{in}) d\tau_{in} & i = 1 \\ \int_{s_{i-1}}^{s_i} f(\tau_{in}) d\tau_{in} & i = 2, \dots, I-1 \\ \int_{s_{I-1}}^{+\infty} f(\tau_{in}) d\tau_{in} & i = I \end{cases} \tag{3.12}$$

where $f(\tau_{in})$ is the Gaussian PDF of τ_{in}. An example of discretization is given in FIGURE 3.7.

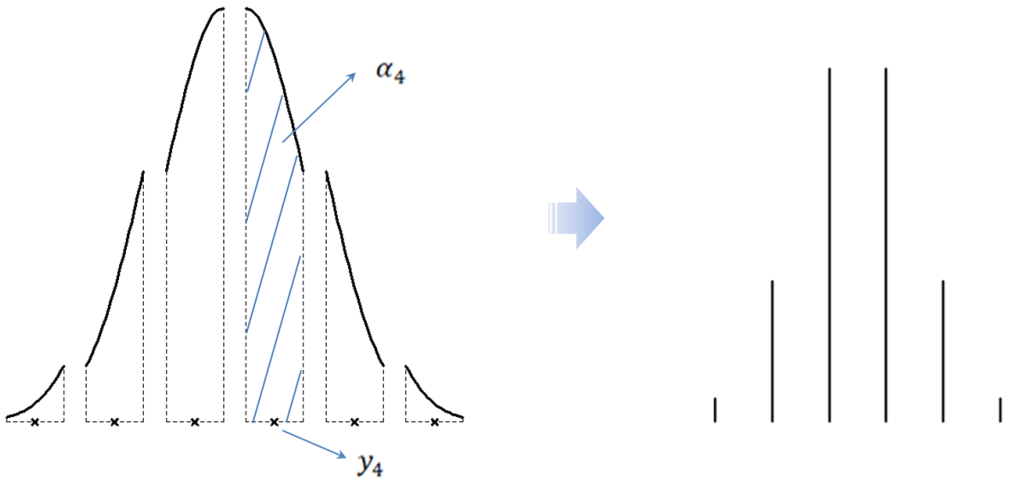


FIGURE 3.7 Discretization of $N(\mu_{\tau_{in}},\ \sigma_{\tau_{in}}^2)$ setting $I = 6$

After the discretization, we compute $\mu_{\tau_{out}}$, $\sigma_{\tau_{out}}^2$, μ_{gd} and σ_{gd}^2 by:

$$\begin{cases} \mu_{\tau_{out}} = \sum_{i=1}^{I} \alpha_i \cdot E(\tau_{out} | y_i, c) \\ \sigma_{\tau_{out}}^2 = \sum_{i=1}^{I} \alpha_i \cdot \left\{ Var(\tau_{out} | y_i, c) + \left[E(\tau_{out} | y_i, c) - \mu_{\tau_{out}} \right]^2 \right\} \end{cases} \tag{3.13}$$

$$\begin{cases} \mu_{gd} = \sum_{i=1}^{I} \alpha_i \cdot E(gd|y_i, c) \\ \sigma_{gd}^2 = \sum_{i=1}^{I} \alpha_i \cdot \left\{ Var(gd|y_i, c) + \left[E(gd|y_i, c) - \mu_{gd} \right]^2 \right\} \end{cases} \tag{3.14}$$

3.3.3 Continuous version

The discrete version in EQUATIONS (3.13) – (3.14) requires an additional Gaussian assumption plus a step of discretization, which increases the CPU time. In this section, we present another version derived from EQUATION (3.8), and compare these two versions of moments propagation.

As the discrete version, X and Y in EQUATION (3.8) are replaced respectively by τ_{in} and τ_{out}. Then, to compute the integrals, we suppose that in certain interval conditional moments $E(\tau_{out}|\tau_{in}, c)$ and $Var(\tau_{out}|\tau_{in}, c)$ depend linearly on τ_{in}, as:

$$\begin{cases} E(\tau_{out}|\tau_{in}, c) = b_1 + b_2 \cdot \tau_{in} \\ Var(\tau_{out}|\tau_{in}, c) = b_3 + b_4 \cdot \tau_{in} \end{cases} \tag{3.15}$$

where b_1, b_2, b_3, b_4 are values to be identified. The assumed relationships in EQUATION (3.15) are reasonable, because the interpolation techniques presented in SECTION 3.3.1 are based on the assumption that in any interval $[\tau_m, \tau_{m+1}]$ conditional moments are linear in τ_{in}.

Combining EQUATION (3.8) with (3.15), we can compute $\mu_{\tau_{out}}$, $\sigma_{\tau_{out}}^2$:

$$\begin{aligned} \mu_{\tau_{out}} &= \int E(\tau_{out}|\tau_{in}, c) \cdot f(\tau_{in}) d\tau_{in} \\ &= b_1 + b_2 \cdot \int \tau_{in} \cdot f(\tau_{in}) d\tau_{in} \\ &= b_1 + b_2 \cdot \mu_{\tau_{in}} \end{aligned} \tag{3.16}$$

$$\sigma_{\tau_{out}}^2 = \int \left\{ Var(\tau_{out}|\tau_{in}, c) + \left[E(\tau_{out}|\tau_{in}, c) - \mu_{\tau_{out}} \right]^2 \right\} \cdot f(\tau_{in}) d\tau_{in}$$

$$= \int \left[b_3 + b_4 \cdot \tau_{in} + \left(b_2 \cdot \tau_{in} - b_2 \cdot \mu_{\tau_{in}} \right)^2 \right] \cdot f(\tau_{in}) d\tau_{in}$$

$$= \left(b_3 + b_4 \cdot \int \tau_{in} \cdot f(\tau_{in}) d\tau_{in} \right) + \int \left(b_2 \cdot \tau_{in} - b_2 \cdot \mu_{\tau_{in}} \right)^2 \cdot f(\tau_{in}) d\tau_{in}$$

$$= \left(b_3 + b_4 \cdot \mu_{\tau_{in}} \right) + \left(b_2 \cdot \sigma_{\tau_{in}} \right)^2 \quad (3.17)$$

where $f(\tau_{in})$ is the PDF of τ_{in}. Note that in EQUATIONS (3.16) – (3.17), $f(\tau_{in})$ is not explicitly known, while $\mu_{\tau_{in}}$ and $\sigma_{\tau_{in}}^2$ are required.

Typically, suppose $\mu_{\tau_{in}} \in (\tau_6, \tau_7)$ and $c \in (c_2, c_3)$, then according to the bilinear interpolation techniques, we have:

$$E(\tau_{out}|\tau_{in}, c) \approx \frac{\tau_7 - \tau_{in}}{\tau_7 - \tau_6} \cdot E(\tau_{out}|\tau_6, c) + \frac{\tau_{in} - \tau_6}{\tau_7 - \tau_6} \cdot E(\tau_{out}|\tau_7, c)$$

$$= \frac{\tau_7 \cdot E(\tau_{out}|\tau_6, c) - \tau_6 \cdot E(\tau_{out}|\tau_7, c)}{\tau_7 - \tau_6} + \frac{E(\tau_{out}|\tau_7, c) - E(\tau_{out}|\tau_6, c)}{\tau_7 - \tau_6} \cdot \tau_{in} \quad (3.18)$$

$$Var(\tau_{out}|\tau_{in}, c) \approx \frac{\tau_7 - \tau_{in}}{\tau_7 - \tau_6} \cdot Var(\tau_{out}|\tau_6, c) + \frac{\tau_{in} - \tau_6}{\tau_7 - \tau_6} \cdot Var(\tau_{out}|\tau_7, c)$$

$$= \frac{\tau_7 \cdot Var(\tau_{out}|\tau_6, c) - \tau_6 \cdot Var(\tau_{out}|\tau_7, c)}{\tau_7 - \tau_6} + \frac{Var(\tau_{out}|\tau_7, c) - Var(\tau_{out}|\tau_6, c)}{\tau_7 - \tau_6} \cdot \tau_{in} \quad (3.19)$$

Combining EQUATION (3.15) with (3.18) – (3.19), we identify b_1, b_2, b_3, b_4:

$$\begin{cases} b_1 = \dfrac{\tau_7 \cdot E(\tau_{out}|\tau_6, c) - \tau_6 \cdot E(\tau_{out}|\tau_7, c)}{\tau_7 - \tau_6} \\ b_2 = \dfrac{E(\tau_{out}|\tau_7, c) - E(\tau_{out}|\tau_6, c)}{\tau_7 - \tau_6} \\ b_3 = \dfrac{\tau_7 \cdot Var(\tau_{out}|\tau_6, c) - \tau_6 \cdot Var(\tau_{out}|\tau_7, c)}{\tau_7 - \tau_6} \\ b_4 = \dfrac{Var(\tau_{out}|\tau_7, c) - Var(\tau_{out}|\tau_6, c)}{\tau_7 - \tau_6} \end{cases} \quad (3.20)$$

Similarly, μ_{gd} and σ_{gd}^2 are computed by replacing the conditional moments of τ_{out} with those of gd in EQUATIONS (3.16) – (3.17) and (3.20).

In TABLE 3.2, we compare the accuracy of the discrete and continuous propagation techniques. Under diverse conditions, like different input slope mean $\mu_{\tau_{in}}$ and variance $\sigma_{\tau_{in}}^2$, the standard deviations are computed respectively by the two versions of techniques, denoted as $\hat{\sigma}_{\tau_{out}}$ and $\hat{\sigma}_{gd}$; the results from MC simulation are considered as "golden values", denoted as $\sigma_{\tau_{out}}$ and σ_{gd}. The errors in TABLE 3.2 are the average values of 200 different cases. Considering accuracy and computational cost, especially the additional step of discretization for the discrete version, it seems appropriate to propagate moments using the continuous version.

TABLE 3.2 Comparison of discrete and continuous propagation techniques (65 nm)

		$\frac{\hat{\sigma}_{\tau_{out}} - \sigma_{\tau_{out}}}{\sigma_{\tau_{out}}}\%$		$\frac{\hat{\sigma}_{gd} - \sigma_{gd}}{\sigma_{gd}}\%$	
		INV	*NOR*	*INV*	*NOR*
discrete	$I = 4$	5.7%	5.4%	2.3%	4.0%
	$I = 6$	5.1%	4.3%	1.6%	2.9%
	$I = 8$	4.9%	3.9%	1.4%	2.6%
continuous		4.7%	3.7%	1.0%	2.3%

3.4 PATH DELAY DISTRIBUTION

For a timing path of K cells, if the moments propagation technique allows iteratively computing cell delay moments $\mu_{gd_k}, \sigma_{gd_k}^2, (k = 1, 2, \dots, K)$, then the path delay pd, which is the sum of all cell delays, has the mean and variance given by:

$$\begin{cases} \mu_{pd} = \sum_{k=1}^{K} \mu_{gd_k} \\ \sigma_{pd}^2 = \sum_{k=1}^{K} \sum_{m=1}^{K} \rho_{km} \cdot \sigma_{gd_k} \sigma_{gd_m} \end{cases} \tag{3.21}$$

where ρ_{km} is the correlation $cor(gd_k, gd_m)$.

In probability theory, the ***central limit theorem*** states conditions under which the sum of a sufficiently large number of independent random variables, each with finite mean and variance, will be approximately Gaussian distributed. Even though gd_k and $gd_m, (k \neq m)$ are not independent, it is reasonable to assume that path delay is a Gaussian random variable. Thus, to get the distribution $N(\mu_{pd}, \sigma_{pd}^2)$, according to EQUATION (3.21), all that remains is to estimate the cell-to-cell delay correlation ρ_{km}, which is the topic of the next section.

3.5 ESTIMATION OF DELAY CORRELATION

Delay correlation is one of the most difficult problems in SSTA. This is because cell delay depends in a complex manner on a number of factors, which makes complex the computation of delay correlation as well. In this section, we introduce a technique to estimate delay correlation.

3.5.1 Cell-to-cell delay correlation

A common way to estimate ***Cell-to-cell Delay Correlation*** (CDC) is to approximate the dependency of cell delay on process parameters with a Taylor expansion, and then to translate the correlation between process parameters into correlation between cell delays. For example, setting the number of process parameters to $L = 2$, delay of cell k is modeled as:

$$gd_k \approx gd_{nom,k} + a_{1k} \cdot \Delta p_{1k} + a_{2k} \cdot \Delta p_{2k} \tag{3.22}$$

Then the CDC between gd_1 and gd_2 is computed by:

$$cor(gd_1, gd_2) = \frac{cov(gd_1, gd_2)}{\sigma_{gd_1}\sigma_{gd_2}} \tag{3.23}$$

where $cov(gd_1, gd_2)$ is the covariance between gd_1 and gd_2. Suppose that Δp_{1k} and Δp_{2k} are independent, then we have:

$$\begin{aligned} cov(gd_1, gd_2) &= cov(a_{11} \cdot \Delta p_{11}, a_{12} \cdot \Delta p_{12}) + cov(a_{21} \cdot \Delta p_{21}, a_{22} \cdot \Delta p_{22}) \\ &= a_{11}a_{12} \cdot cov(\Delta p_{11}, \Delta p_{12}) + a_{21}a_{22} \cdot cov(\Delta p_{21}, \Delta p_{22}) \end{aligned} \tag{3.24}$$

If $\Delta p_{lk}, (l = 1,2)$ are further divided into independent inter-die and intra-die component as:

$$\Delta p_{lk} = \Delta p_{inter\ ,lk} + \Delta p_{intra\ ,lk} \qquad (l = 1,2) \tag{3.25}$$

then $cov(\Delta p_{l1}, \Delta p_{l2}), (l = 1,2)$ in EQUATION (3.24) are given by:

$$cov(\Delta p_{l1}, \Delta p_{l2}) = \sigma_{\Delta p_{inter\ ,l1}} \sigma_{\Delta p_{inter\ ,l2}} \cdot cor\left(\Delta p_{inter\ ,l1}, \Delta p_{inter\ ,l2}\right) +$$

$$\sigma_{\Delta p_{intra\ ,l1}} \sigma_{\Delta p_{intra\ ,l2}} \cdot cor\left(\Delta p_{intra\ ,l1}, \Delta p_{intra\ ,l2}\right) \qquad (l = 1,2) \tag{3.26}$$

Given a statistical process model, $cor(gd_1, gd_2)$ is computed by combining EQUATIONS (3.23) – (3.24) and (3.26).

The above technique of computation explains CDC in terms of correlation between process parameters. Theoretically, apart from process parameters, all factors that affect cell delay, like cell type, output load, etc., should be considered. TABLE 3.3 demonstrates that CDC varies with cell type (INV, OR, BUF), output load ($1fF$, $10fF$, $100fF$) and I/O edge (R/F, F/R, R/R, F/F). In this table, the CDC coefficients are estimated with data from MC simulations. As shown in TABLE 3.3, the effects of cell type and I/O edge on CDC are obvious. In addition, it seems that coefficients are brought down by increasing output load.

TABLE 3.3 CDCs varying with cell type, output load and I/O edge (130 nm, 1500 runs)

			INV		BUF	
			$10fF$	$10fF$	$10fF$	$10fF$
			R/F	F/R	R/R	F/F
INV	$1fF$	R/F	0.97	0.73	0.90	0.94
		F/R	0.61	0.97	0.94	0.92
	$10fF$	R/F	0.99	0.76	0.91	0.95
		F/R	0.66	0.99	0.95	0.92
	$100fF$	R/F	0.98	0.62	0.89	0.88
		F/R	0.64	0.99	0.89	0.87
OR	$1fF$	R/R	0.84	0.65	0.87	0.89
		F/F	0.62	0.98	0.91	0.76
	$10fF$	R/R	0.71	0.55	0.76	0.83
		F/F	0.54	0.96	0.84	0.62
	$100fF$	R/R	0.62	0.49	0.64	0.78
		F/F	0.38	0.93	0.77	0.52

As cell delay depends on a number of factors, which affects CDC as well, we propose a technique to compute directly CDC, which avoids handling complex relationship between process parameters. Suppose that process parameters $p_1, p_2, \dots, p_L$ are classified into three groups:

$$\begin{cases} P^{NM} = \left(p_1^{NM}, p_2^{NM}, \dots, p_{n_1}^{NM}\right) \\ P^{PM} = \left(p_1^{PM}, p_2^{PM}, \dots, p_{n_2}^{PM}\right) \\ P^{S} = \left(p_1^{S}, p_2^{S}, \dots, p_{n_3}^{S}\right) \end{cases} \qquad L = n_1 + n_2 + n_3 \tag{3.27}$$

where P^{NM} comprises process parameters characterizing only N-transistors; P^{PM} is the group only related P-transistors; and the parameters of P^S describe both N- and P-transistors. Corresponding to this classification, cell delay is modeled as:

$$gd \approx gd^{NM} + gd^{PM} + gd^{S} \tag{3.28}$$

where gd^{NM}, gd^{PM} and gd^S, according to EQUATION (1.1), are defined by:

$$\begin{cases} gd^{NM} = f_{type,pin,edge}\left(P^{NM}, T, V_{dd}, \tau_{in}, C_{out}\right) \\ gd^{PM} = f_{type,pin,edge}\left(P^{PM}, T, V_{dd}, \tau_{in}, C_{out}\right) \\ gd^{S} = f_{type,pin,edge}\left(P^{S}, T, V_{dd}, \tau_{in}, C_{out}\right) \end{cases} \tag{3.29}$$

As stated in SECTION 3.1.1, for any $l_1 \neq l_2, (l_1, l_2 = 1, 2, \dots L)$, p_{l_1} and p_{l_2} are independent. Thus, it is reasonable to assume:

$$\begin{cases} cor(gd^{NM}, gd^{PM}) = 0 \\ cor(gd^{NM}, gd^{S}) = 0 \\ cor(gd^{PM}, gd^{S}) = 0 \end{cases} \tag{3.30}$$

With the assumptions above and the cell delay model in EQUATION (3.28), CDC between gd_k and gd_m can then be computed according to:

$$\rho_{km} = \frac{cov(gd_k, gd_m)}{\sigma_{gd_k}\sigma_{gd_m}} \tag{3.31}$$

where

$$cov(gd_k, gd_m) = cov(gd_k^{NM} + gd_k^{PM} + gd_k^{S}, gd_m^{NM} + gd_m^{PM} + gd_m^{S})$$
$$= cov(gd_k^{NM}, gd_m^{NM}) + cov(gd_k^{NM}, gd_m^{PM}) + cov(gd_k^{NM}, gd_m^{S}) +$$
$$cov(gd_k^{PM}, gd_m^{NM}) + cov(gd_k^{PM}, gd_m^{PM}) + cov(gd_k^{PM}, gd_m^{S}) +$$
$$cov(gd_k^{S}, gd_m^{NM}) + cov(gd_k^{S}, gd_m^{PM}) + cov(gd_k^{S}, gd_m^{S})$$
$$= cov(gd_k^{NM}, gd_m^{NM}) + cov(gd_k^{PM}, gd_m^{PM}) + cov(gd_k^{S}, gd_m^{S}) \quad (3.32)$$

In EQUATION (3.1), the variations of each process parameter p_l are divided into a inter-die component $\Delta p_{inter,l}$ and a intra-die component $\Delta p_{intra,l}$, which are independent of each other. Similarly, we decompose gd^{NM} with independent inter-die and intra-die components as:

$$gd^{NM} = gd_{inter}^{NM} + gd_{intra}^{NM} \quad (3.33)$$

Adding the approximation below:

$$cor\left(gd_{k,inter}^{NM}, gd_{m,inter}^{NM}\right) \approx 1 \quad (3.34)$$

and knowing that $cor\left(gd_{k,intra}^{NM}, gd_{m,intra}^{NM}\right) = 0$, then the covariance $cov(gd_k^{NM}, gd_m^{NM})$ is computed by:

$$cov(gd_k^{NM}, gd_m^{NM}) = cov\left(gd_{k,inter}^{NM}, gd_{m,inter}^{NM}\right) + cov\left(gd_{k,inter}^{NM}, gd_{m,intra}^{NM}\right) +$$
$$cov\left(gd_{k,intra}^{NM}, gd_{m,inter}^{NM}\right) + cov\left(gd_{k,intra}^{NM}, gd_{m,intra}^{NM}\right)$$
$$\approx \sigma_{gd_k,inter}^{NM} \cdot \sigma_{gd_m,inter}^{NM} \quad (3.35)$$

Another two terms $cov(gd_k^{PM}, gd_m^{PM})$ and $cov(gd_k^{S}, gd_m^{S})$ in EQUATION (3.32) are obtained in a similar way. Finally, we have:

$$cov(gd_k, gd_m) \approx \sigma_{gd_k,inter}^{NM} \cdot \sigma_{gd_m,inter}^{NM} + \sigma_{gd_k,inter}^{PM} \cdot \sigma_{gd_m,inter}^{PM} +$$
$$\sigma_{gd_k,inter}^{S} \cdot \sigma_{gd_m,inter}^{S} \quad (3.36)$$

From EQUATION (3.36), an immediate drawback of the technique appears: $\sigma_{gd,inter}^{NM}$, $\sigma_{gd,inter}^{PM}$, $\sigma_{gd,inter}^{S}$ of each cell must be characterized. This additional information requires a lot of CPU time when constructing the statistical timing library.

3.5.2 Path-to-path delay correlation

To compute circuit delay with the algorithms in [46] based on EQUATIONS (2.10) – (2.11), ***Path-to-path Delay Correlation*** (PDC) is required. Suppose two paths constituted respectively by K_1 and K_2 cells, then the PDC is computed by:

$$cor(pd_1, pd_2) = \frac{cov(pd_1, pd_2)}{\sigma_{pd_1}\sigma_{pd_2}} \tag{3.37}$$

Adopting the setting of SECTION 3.5.1 to compute CDC, we have:

$$cov(pd_1, pd_2) = cov\left(\sum_{k_1=1}^{K_1} gd_{k_1}, \sum_{k_2=1}^{K_2} gd_{k_2}\right) = \sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2} cov\left(gd_{k_1}, gd_{k_2}\right) \tag{3.38}$$

If the two paths have common cells, i.e. k_1 and k_2 indicate the same cell, then:

$$cov\left(gd_{k_1}, gd_{k_2}\right) = \sigma_{gd_{k_1}} \cdot \sigma_{gd_{k_2}} \tag{3.39}$$

otherwise:

$$cov\left(gd_{k_1}, gd_{k_2}\right) \approx \sigma_{gd_{k_1},inter}^{NM} \cdot \sigma_{gd_{k_2},inter}^{NM} + \sigma_{gd_{k_1},inter}^{PM} \cdot \sigma_{gd_{k_2},inter}^{PM} + \sigma_{gd_{k_1},inter}^{S} \cdot \sigma_{gd_{k_2},inter}^{S} \tag{3.40}$$

3.6 VALIDATION AND DISCUSSION

In this section, our SSTA engine is validated by comparing its results with those from MC simulations. Next, the advantages and the computational cost of the engine are presented. Finally, we discuss some ideas to improve the engine.

3.6.1 Validation

As presented in SECTION 3.1, we first characterized conditional moments of timing variables, and then constructed the statistical timing library. In the second step, a certain number of critical paths were extracted from the considered circuits using CTA under the software RTL Compiler. Then, we performed SSTA with the timing engine implemented by the statistical computing and graphic tool R [49]. Finally, we ran MC simulations for comparison.

As shown in TABLES 3.4 – 3.5, the validation is done respectively in the 130 nm and 65 nm technology. The three considered circuits are b01, b05 and b07 of the ITC99 benchmark. In these two tables, relative errors on estimated means and standard deviations of path delays are respectively less than 5% and 10%. These errors are acceptable in the context of timing analysis. Moreover, most of the standard deviations are a little overestimated, which will reduce the probability of violating the setup and hold time constraints if ICs are designed with this SSTA framework.

TABLE 3.4 Validation in the 130 nm technology

name	path	logical depth	path delay (ps)				error (%)	
			MC simulations (1500 runs)		SSTA (continuous version)		$\frac{\hat{\mu}_{pd}-\mu_{pd}}{\mu_{pd}}$%	$\frac{\hat{\sigma}_{pd}-\sigma_{pd}}{\sigma_{pd}}$%
			μ_{pd}	σ_{pd}	$\hat{\mu}_{pd}$	$\hat{\sigma}_{pd}$		
b01	1	5	665.5	40.0	690.5	42.7	3.8%	6.3%
	2	6	590.7	34.9	605.8	36.4	2.6%	4.1%
	3	7	598.3	35.3	610.3	35.5	2.0%	0.6%
	4	5	660.3	39.4	680.2	42.1	3.0%	6.4%
	5	6	644.1	38.7	658.9	41.0	2.3%	5.6%
b05	1	13	1185.8	70.2	1206.6	71.8	1.8%	2.2%
	2	17	1106.4	66.5	1098.6	67.1	−0.7%	0.9%
	3	18	991.3	61.7	1027.0	65.3	3.6%	5.5%
	4	20	1249.3	75.8	1242.7	75.7	−0.5%	−0.1%
	5	19	1294.6	78.2	1291.9	78.5	−0.2%	0.4%
b07	1	10	722.0	44.2	725.2	44.4	0.4%	0.5%
	2	10	738.7	45.3	750.6	46.3	1.6%	2.2%
	3	9	720.2	43.9	727.4	46.1	1.0%	4.8%
	4	11	740.5	45.6	735.4	47.1	−0.7%	3.2%
	5	9	722.1	44.2	730.1	46.3	1.1%	4.5%

TABLE 3.5 Validation in the 65 nm technology

name	path	logical depth	path delay (ps)				error (%)	
			MC simulations (1500 runs)		SSTA (continuous version)		$\frac{\hat{\mu}_{pd}-\mu_{pd}}{\mu_{pd}}$%	$\frac{\hat{\sigma}_{pd}-\sigma_{pd}}{\sigma_{pd}}$%
			μ_{pd}	σ_{pd}	$\hat{\mu}_{pd}$	$\hat{\sigma}_{pd}$		
b01	1	9	500.2	26.8	489.7	27.5	−2.1%	2.5%
	2	8	495.0	27.4	492.6	27.9	−0.5%	1.8%
	3	7	461.9	25.9	460.2	26.4	−0.4%	2.1%
	4	9	496.2	27.1	498.5	27.4	0.5%	1.1%
	5	7	475.7	26.2	475.8	27.1	0.0%	3.7%
b05	1	25	1067.6	57.8	1050.2	57.3	−1.6%	−0.8%
	2	23	1077.7	58.5	1069.5	59.9	−0.8%	2.5%
	3	22	1080.9	57.7	1073.7	59.5	−0.7%	3.1%
	4	22	1110.4	59.0	1113.2	60.4	0.3%	2.3%
	5	23	1077.7	58.5	1069.5	59.9	−0.8%	2.5%
b07	1	12	657.4	34.3	642.2	34.9	−2.3%	1.7%
	2	11	657.0	34.2	656.7	36.2	−0.1%	5.8%
	3	13	659.1	35.2	647.6	36.1	−1.8%	2.6%
	4	10	674.9	36.2	685.9	38.3	1.6%	5.8%
	5	11	654.4	35.6	655.2	38.0	0.1%	6.8%

The validation of the techniques to compute CDCs and PDCs is given respectively in FIGURES 3.8 and 3.9. Denote correlation coefficients computed by the SSTA engine as $\hat{\rho}$. Considering coefficients ρ from MC simulations as reference, the absolute errors e_{abs} and relative errors e_{rel} are computed by:

$$\begin{cases} e_{abs} = |\hat{\rho} - \rho| \\ e_{rel} = \dfrac{|\hat{\rho} - \rho|}{\rho}\% \end{cases} \tag{3.41}$$

In FIGURE 3.8, the majority of points are in the region with $e_{abs} \leq 0.2$. In addition, most of the points outside this dashed region, i.e. $e_{abs} > 0.2$, are overestimated, which lead to overestimation on path delay standard deviations according to EQUATION (3.21). This explains why, in TABLE 3.5, all relative errors of path delay standard deviations expect for path 1 of circuit b05 are positive. In FIGURE 3.9, all points are in the region with $e_{rel} \leq 20\%$. Thus, the accuracy of PDCs computed by the SSTA engine is better than that of CDCs.

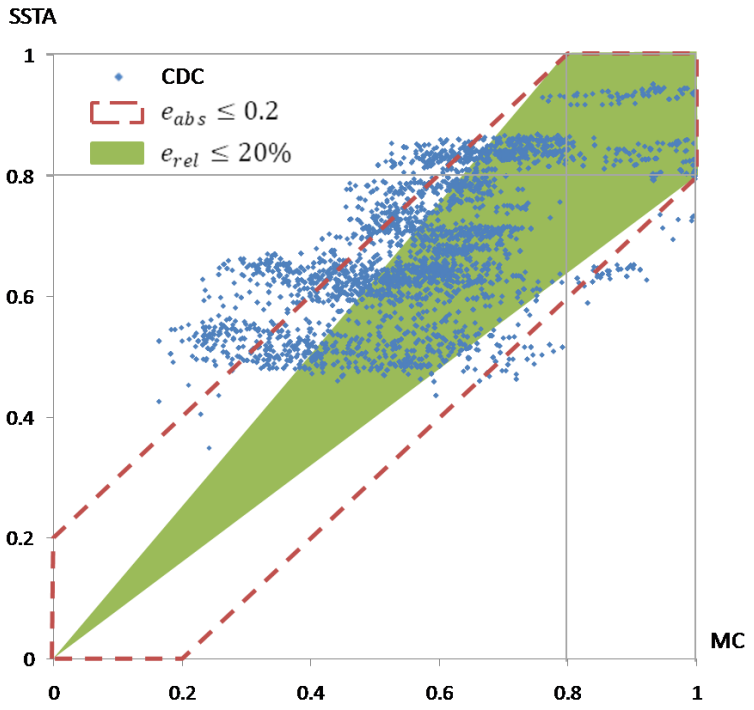


FIGURE 3.8 Validation of the technique to compute CDCs (65 nm)

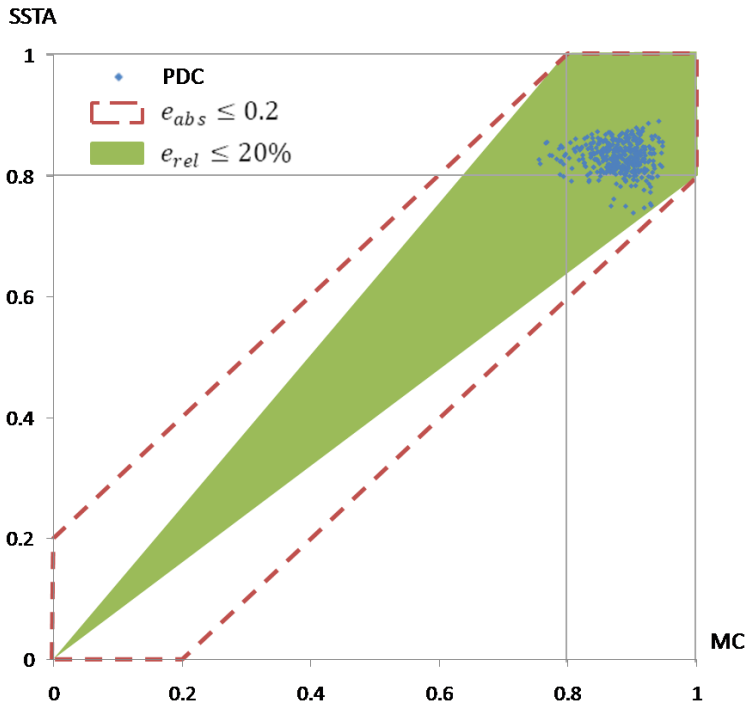


FIGURE 3.9 Validation of the technique to compute PDCs (65 nm)

In order to obtain detailed information, we compute the following proportions:

$$\begin{cases} p_{oe} = \frac{1}{N} \cdot \sum_{i=1}^{N} if\{\hat{\rho}_i - \rho_i < 0, then\ 1, else\ 0\} \\ p_{ue} = \frac{1}{N} \cdot \sum_{i=1}^{N} if\{\hat{\rho}_i - \rho_i > 0, then\ 1, else\ 0\} \\ p_{abs} = \frac{1}{N} \cdot \sum_{i=1}^{N} if\{e_{abs} \le 0.2, then\ 1, else\ 0\} \\ p_{rel} = \frac{1}{N} \cdot \sum_{i=1}^{N} if\{e_{rel} \le 20\%, then\ 1, else\ 0\} \end{cases} \tag{3.42}$$

where N is the sample size, and $p_{oe}, p_{ue}, p_{abs}, p_{rel}$ represent respectively the proportion of points overestimated, underestimated, in the region $e_{abs} \le 0.2$ and $e_{rel} \le 20\%$. TABLE 3.6 gives these proportions in percentage. The most important information in this table is that 81% of the CDCs are overestimated while 84% of the PDCs are underestimated, which is the expected results.

TABLE 3.6 Information about the accuracy of computed CDCs and PDCs

	N	p_{oe}%	p_{ue}%	p_{abs}%	p_{rel}%
CDC	10532	81%	19%	76%	55%
PDC	780	16%	84%	100%	100%

To explain why overestimate of CDC and underestimate of PDC are preferable, we return to the problem of timing verification. For ease of description, consider the setup time constraint below:

$$pd_{data} - pd_{clk} < T_{CLK} \tag{3.43}$$

where T_{CLK} is the clock period and pd_{data}, pd_{clk} are respectively delays of data path and clock path. Comparing with EQUATION (1.3), pd_{data} corresponds to the left hand side and pd_{clk} to the first two terms of the right hand side. Note that T_{CLK} is a constant while pd_{data}, pd_{clk} are random variables. Then, we have:

$$Var(pd_{data} - pd_{clk}) = \sigma_{data}^2 + \sigma_{clk}^2 - 2 \cdot \rho_{dc} \cdot \sigma_{data} \cdot \sigma_{clk} \tag{3.44}$$

where σ_{data}^2, σ_{clk}^2 are respectively the variance of pd_{data} and pd_{clk}; ρ_{dc} is the PDC between pd_{data} and pd_{clk} and varies in the interval [0,1].

According to EQUATION (3.44), $Var(pd_{data} - pd_{clk})$ will increase if one or more of the following cases appears:

a) σ_{data} increases;
b) σ_{clk} increases;
c) ρ_{dc} decreases.

In other words, both the overestimation of CDC and the underestimation of PDC result in the overestimation of $Var(pd_{data} - pd_{clk})$. As illustrated in FIGURE 3.10, if the distribution with overestimated variance satisfies the setup time constraint, then so will the actual distribution. Thus, this overestimation of $Var(pd_{data} - pd_{clk})$, i.e. overestimate of CDC and underestimate of PDC, is a little conservative but preferable, which validates the techniques of computing delay correlations.

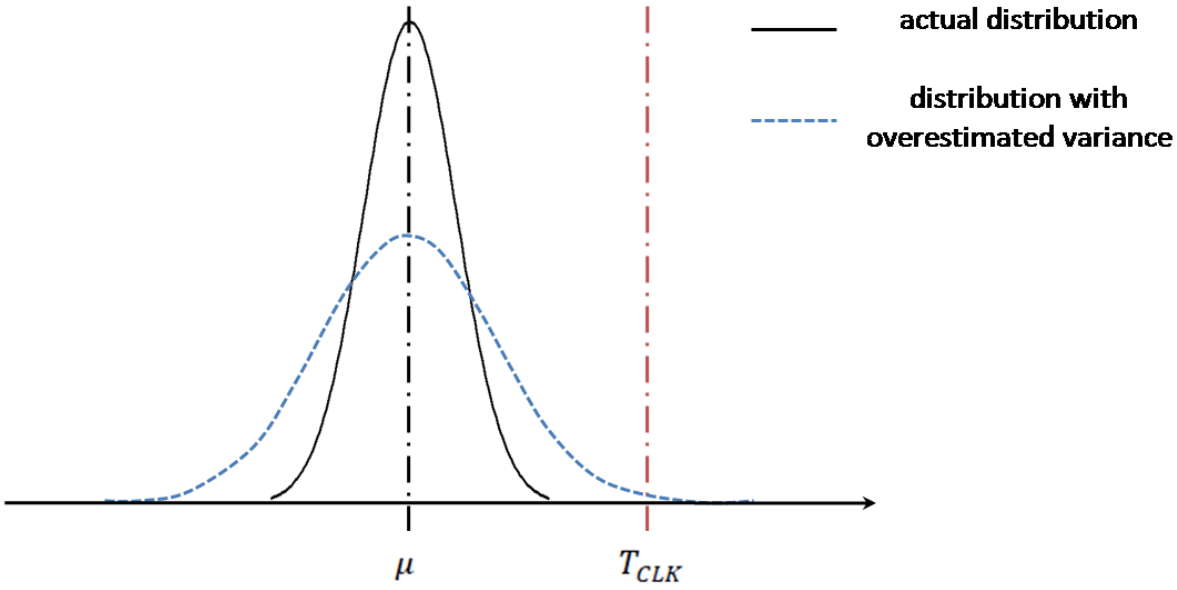


FIGURE 3.10 Illustration of preferable overestimation on $Var(pd_{data} - pd_{clk})$

3.6.2 Quality of the SSTA engine

Accuracy and computational cost are the two most important criteria to evaluate the quality of a SSTA method. In SECTION 3.6.1, an overview on the accuracy of the proposed SSTA engine

has been given. In this section, we turn our attention to its computational cost. TABLE 3.7(a) gives some examples of CPU time gains of the SSTA engine compared to MC simulations. In this table, to compute delay distribution of the same path, the SSTA engine implemented by the statistical computing software R [49] is over 10^5 times faster than MC simulations ran under HSPICE [43]. TABLE 3.7(b) gives the running environments of these two methods.

TABLE 3.7 Computational cost of MC simulations and our SSTA engine

(a) Some comparisons of computational cost

path	logical depth	CPU time (s)		st/et (simulation time : st SSTA time : et)
		MC simulations (1500 runs)	SSTA (continuous version)	
1	5	2794.02	0.02	1.40×10^5
2	10	5245.12	0.03	1.75×10^5
3	15	6914.28	0.06	1.15×10^5
4	20	9881.50	0.08	1.24×10^5
5	25	12020.70	0.11	1.09×10^5

(b) Running environments of MC simulations and the SSTA engine

	platform	CPU	number of CPU	CPU frequency	memory	software
MC simulations	Unix	Ultra SPARC III	8	900MHz	32G	HSPICE
SSTA	Windows	Intel Pentium D	1	2800MHz	1G	R

TABLE 3.8 shows the influences of CDCs and slope variations on the accuracy of estimated path delay standard deviations. Apart from the technique to compute CDCs in EQUATIONS (3.31) and (3.36), the following two extreme cases are considered:

- CDCs $\rho_{k_1 k_2}$ are set to "1" for any k_1, k_2;
- CDCs $\rho_{k_1 k_2}$ are set to "0" except for $k_1 = k_2$.

As shown in TABLE 3.8, the two extreme cases above respectively lead to an average relative error 23.7% and -56.3%. The last column gives -2.7% if slope variations are not taken into account, which gives a difference of about 8% (i.e. $|-2.7\% - 5.0\%|$) compared to the average using EQUATIONS (3.31) and (3.36). In other words, slope variations should not be neglected, otherwise about 8% of standard deviation is lost.

TABLE 3.8 Influences of CDCs and slope variations

path	logical depth	σ_{pd} (ps)	$\frac{\hat{\sigma}_{pd}-\sigma_{pd}}{\sigma_{pd}}\%$			
			$\rho_{k_1k_2}$ computed by EQUATIONS (3.31) and (3.36)	$\forall k_1, k_2$ $\rho_{k_1k_2}=1$	$\begin{cases}\rho_{k_1k_2}=1, k_1=k_2\\ \rho_{k_1k_2}=0, k_1\neq k_2\end{cases}$	for each cell k, $\sigma^2_{\tau_{in},k}=0$
1	5	22.9	7.0%	23.1%	−33.2%	−3.0%
2	10	36.2	7.7%	25.1%	−50.8%	2.4%
3	15	45.4	0.9%	21.6%	−61.7%	−6.3%
4	20	57.7	6.4%	24.8%	−66.4%	−1.2%
5	25	57.8	2.8%	23.8%	−69.2%	−5.4%
average			5.0%	23.7%	−56.3%	−2.7%

3.6.3 Discussion

The outstanding characteristic of the proposed SSTA engine is the independency of moments propagation on statistical process model and approximation of MAX operation. In other words, if a universally accepted statistical process model appears, the propagation technique could be adapted to still be valid. Moreover, we use the algorithms in [46] based on the linear approximation of MAX to compute circuit delay to date. However, if we improve the engine by also propagating the third moment of cell delays, then the Gaussian assumption on path delay can be changed to, for example, a skew-Normal one. This allows using the skew-Normal based MAX approximation in [32], which would provide better accuracy on computation of circuit delay.

On the downside, the technique to compute delay correlation in SECTION 3.5 depends on statistical process model. What is more important, it cannot take into account correlation between input slopes. These weaknesses should be addressed in the future.

3.7 SUMMARY

The SSTA engine presented in this chapter is implemented by moments propagation, which overcomes some of the weaknesses of existing parametric methods. From the point of view of accuracy, path delay means and standard deviations computed by this engine have relative errors

respectively less than 5% and 10%. The technique to compute delay correlation in general over-estimates CDCs and underestimates PDCs, which is a preferable result. As for CPU time, it is about 10^5 times faster than a 1500 runs MC simulation for the same path.

CHAPTER

4

STATISTICAL TIMING LIBRARY

This chapter introduces techniques to improve the quality of our statistical timing library and to reduce the CPU time of timing characterization. In SECTION 4.1, we present an input signal model derived from the ***Log-Logistic*** *(LL) distribution, and an output load model based on inverters. SECTION 4.2 proposes techniques to reduce dimensions to save CPU time during the procedure of characterization.*

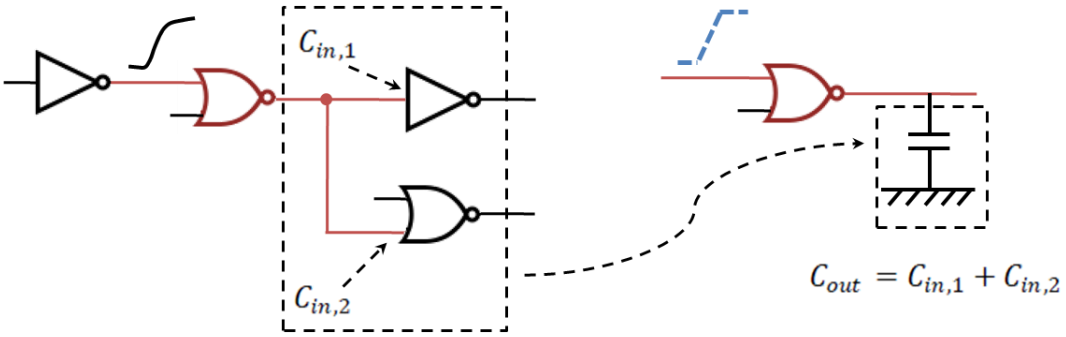
Moments propagation techniques, in which cell delays means and variances are computed, are the kernel of the SSTA engine. The accuracy of this propagation technique is mainly determined by the quality of the statistical timing library. This quality depends on the accuracy of conditional moments, as well as the number of conditional moments contained in each lookup table. As conditional moments are estimated by data from ***Monte Carlo*** (MC) simulations, their accuracy can be improved by increasing the number of runs. In addition, the dependency of conditional moments on input slope and output load are approximated by lookup tables and bilinear interpolations. Thus, increasing the number of lookup values can produce better results.

In fact, to improve the quality of the library, the most crucial technique is to reduce the errors induced by input signals and output load models when doing timing characterization. SECTION 4.1 gives details on this topic.

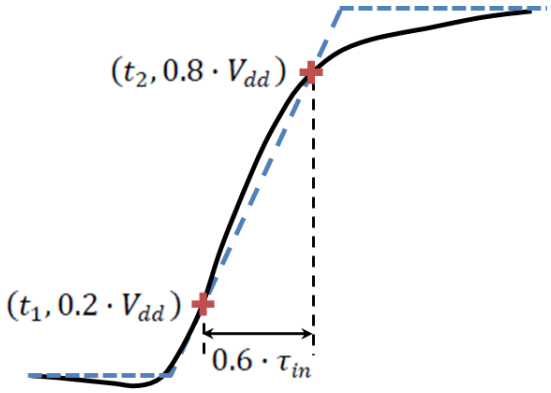
4.1 TIMING CHARACTERIZATION

Timing characterization is the procedure to pre-characterize timing information for each type of cell by MC simulations. In our context, timing information includes conditional moments of cell delay and output slope for diverse combination of factors. Among these factors, cell type, input/output pin, and input/output edge are deterministic; temperature and supply voltage are set to be constants; samples of process parameters are randomly generated; as for input slope and output load, the conventional method is to use a linear ramp model and capacitors, as shown in FIGURE 4.1.

In FIGURE 4.1(a), the left panel is the considered OR cell in a circuit; in the right panel, we extract only the OR cell and its output load is replaced by a capacitor, the charge of which is the sum of charges of all connected pins in the left panel. FIGURE 4.1(b) shows an input signal from circuit simulation and its approximation: linear ramp model. The straight dashed line typically passes through the two points $(t_1, 0.2 \cdot V_{dd})$ and $(t_2, 0.8 \cdot V_{dd})$. Then, the signal is determined by τ_{in} and V_{dd}.



(a) Approximation of output load with capacitor



(b) Linear ramp (dashed line) approximation of input signal (solid line)

FIGURE 4.1 Conventional approximations of input slope and output load

Note that timing characterization is done with standalone cells instead of a complete circuit. This is because we have no information about how a cell would be connected during the procedure of constructing statistical timing library. In addition, the structure of connection is different from one circuit design to another. In consequence, we use input signal and output load models to approximate what could happen at the input and output pins of a cell in real circuits.

This conventional method is simple and efficient. However, as the magnitude of process variations grows, such a method cannot provide acceptable results any more, especially when capturing variations of timing variables. In fact, charges of capacitors are constants during MC simulation, i.e. they do not depend on random process parameters, whereas charges of input pins do

depend on these parameters and, therefore, are random. Thus, conditional variances will be underestimated if characterization is done using capacitors.

In order to increase the quality of approximations, we propose an input signal model based on the ***Log-Logistic*** (LL) ***Cumulative Distribution Function*** (CDF), and use inverters to replace capacitors for output load.

4.1.1 Input signal model

In this section, we only focus on rising edges for simplicity, because rising edges and falling edges are similar in terms of shape.

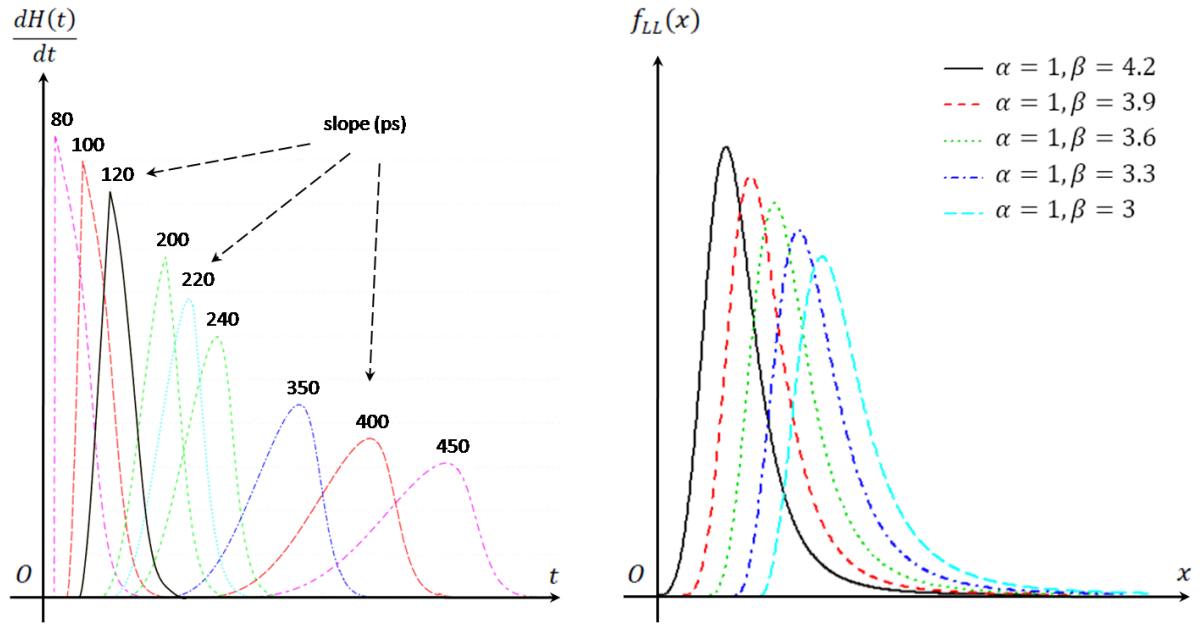
First, we study the input signal characteristics. In the context of digital IC, a signal can be described by a voltage function $H(t)$ depending on time t. FIGURE 4.2(a) gives the derivatives $\frac{dH(t)}{dt}$ of some typical signals of different slopes. From this figure, it is obvious that the linear ramp model, the derivative of which is a constant, is of low accuracy. In FIGURE 4.2(b), LL ***Probability Density Functions*** (PDF) $f_{LL}(x)$ of different parameters are plotted. These PDFs have similar forms to some of the signal derivatives, especially those located on the left part of FIGURE 4.2(a), e.g. signals with slope less than 120 ps in the figure, which in practice occur 80% of times [50]. In addition, if we normalize a signal $H(t)$ by its total amplitude R and transform it to satisfy the condition $\frac{H(t)}{R} \in [0,1]$, then beyond a certain moment t_{min}, $\frac{H(t)}{R}$ looks like a CDF, because:

- it is monotone increasing on $[t_{min}, \infty)$,
- $\frac{H(t_{min})}{R} = 0$ and $lim_{t\to\infty} \frac{H(t)}{R} = 1$.

Therefore, it is feasible to approximate input and output signal functions with LL CDFs:

$$F(x;\alpha,\beta) = \left[\left(\frac{\alpha}{x}\right)^{\beta} + 1\right]^{-1} \qquad (x > 0) \tag{4.1}$$

where $\alpha, \beta > 0$ are two parameters to identify. As shown above, the expression of LL CDFs is simple, which is the main advantage of using LL-based approximation.



(a) Derivatives of some signals

(b) PDFs (derivatives of CDFs) of LL distributions

FIGURE 4.2 Comparison of signals and LL distributions

Before proceeding, the notations for the definition of LL-based models are given in FIGURE 4.3. Note that the part that is below the zero line is a normal electronic phenomenon while the signal is switching. In order to model this special part, a signal is divided into two segments: $t \leq t_{min}$ and $t > t_{min}$.

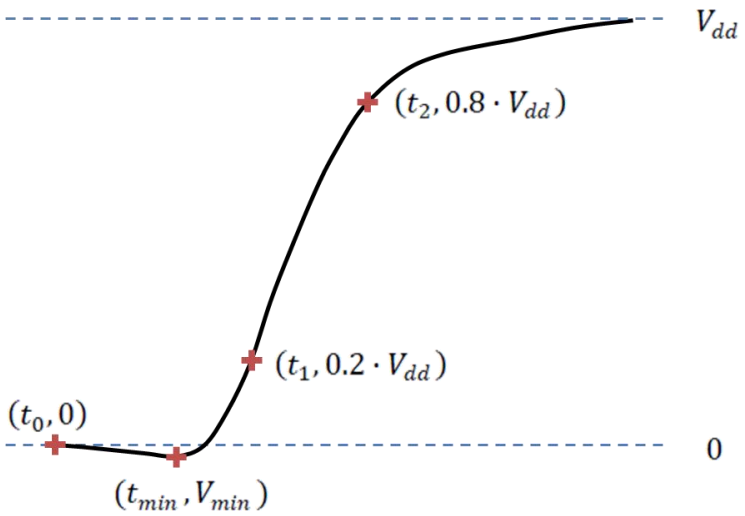


FIGURE 4.3 Notations of input signal model

According to the notations, we define:

$$\begin{cases} \tau_{in} = \dfrac{5}{3} \cdot (t_2 - t_1) \\ \Delta V = |0 - V_{min}| = |V_{min}| \\ \Delta t = t_{min} - t_0 \end{cases} \tag{4.2}$$

Denote $\widehat{H}(t)$ the approximating function. Based on EQUATION (4.1), we have the model below:

$$\hat{V} = \widehat{H}(t) = \begin{cases} -\dfrac{\Delta V}{\Delta t} \cdot (t + \Delta t - t_{min}) & (t \le t_{min}) \\ (V_{dd} + \Delta V) \cdot \left\{ \left[\dfrac{\alpha}{(t - t_{min})/\tau_{in}} \right]^{\beta} + 1 \right\}^{-1} - \Delta V & (t > t_{min}) \end{cases} \tag{4.3}$$

where τ_{in} is known; t_{min} may be any value greater than Δt, and indicates the location of the approximated signal; $\alpha, \beta, \Delta t, \Delta V$ are values to identify.

In FIGURE 4.1(b), the linear ramp is only determined by the input slope τ_{in}. For the LL-based model, the idea is to compute $\alpha, \beta, \Delta t, \Delta V$ from τ_{in} so that the approximated signal is determined by τ_{in} as well. For this purpose, we build functions:

$$\begin{cases} \Delta V = g_{\Delta V}(\tau_{in}) \\ \Delta t = g_{\Delta t}(\tau_{in}) \\ \beta = g_{\beta}(\tau_{in}) \end{cases} \tag{4.4}$$

Once $\Delta V, \Delta t, \beta$ are obtained, then according to EQUATION (4.3), parameter α may be computed with the two points $(t_1, 0.2 \cdot V_{dd})$ and $(t_2, 0.8 \cdot V_{dd})$. To identify the functions in EQUATION (4.4), we follow the three steps below:

a) collect data from MC simulations;
b) analyze the dependency of $\Delta V, \Delta t, \beta$ on τ_{in} by plots, and propose simple explicit functions of $g_{\Delta V}(\tau_{in}), g_{\Delta t}(\tau_{in})$ and $g_{\beta}(\tau_{in})$;
c) estimate parameters of the proposed functions using the ***Least Squares Method*** (LSM) [51].

First of all, we collect from MC simulations, 1000 output signals of different cells under diverse operating conditions, such as temperature, supply voltage, input slope, output load, etc. For each signal, nine points for the segment $(t > t_{min})$ corresponding to $\omega \cdot V_{dd}, (\omega = 0.1, 0.2, \dots 0.9)$, plus $(t_0, 0)$ and (t_{min}, V_{min}), are measured. Next, according to EQUATION (4.2), we compute $\tau_{in}, \Delta V, \Delta t$. Finally, for each signal, we estimate α, β in EQUATION (4.3) with the nine measured points $(t > t_{min})$ using LSM.

In FIGURE 4.4(a), there exists a trend that ΔV decreases along with the increase of τ_{in}; FIGURE 4.4(b) shows a linear increasing trend of Δt on τ_{in}; in FIGURE 4.4(c), β seems to decrease if τ_{in} increases. Thus, we propose the simple functions below:

$$\begin{cases} \Delta V = \dfrac{C_{\Delta V}}{A_{\Delta V} + B_{\Delta V} \cdot \tau_{in}} \\ \Delta t = A_{\Delta t} + B_{\Delta t} \cdot \tau_{in} \\ \beta = \dfrac{C_{\beta}}{A_{\beta} + B_{\beta} \cdot \tau_{in}} + D_{\beta} \end{cases} \tag{4.5}$$

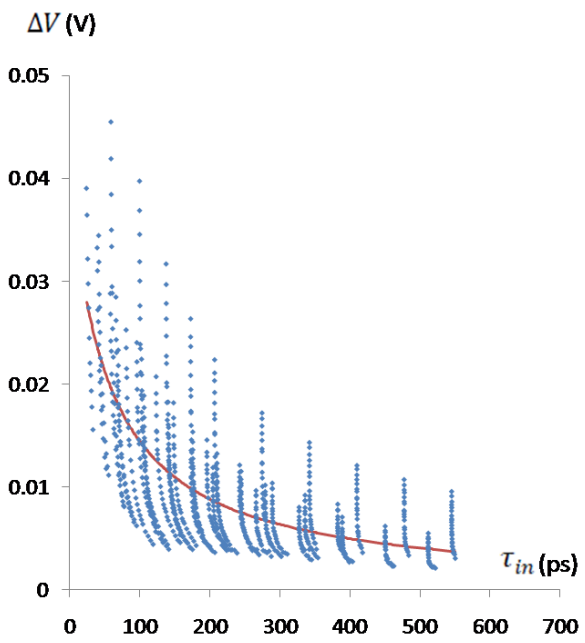
The first two functions are derived from the model of overshoot [52]. The last one comes from FIGURE 4.4(c), which is similar to FIGURE 4.4(a). Using LSM, we have:

$$\begin{cases} A_{\Delta V} = 15.52 \\ B_{\Delta V} = 1.81 \times 10^{11} \\ C_{\Delta V} = 0.45 \end{cases} \qquad \begin{cases} A_{\Delta t} = 4.7 \times 10^{-11} \\ B_{\Delta t} = 0.04 \end{cases} \qquad \begin{cases} A_{\beta} = 0.33 \\ B_{\beta} = 6.9 \times 10^{9} \\ C_{\beta} = 1.32 \\ D_{\beta} = 1.51 \end{cases}$$

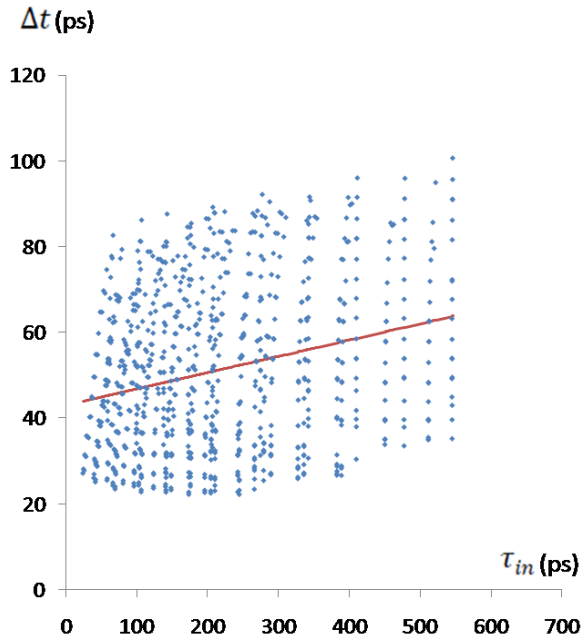
Given an input slope τ_{in}, with EQUATION (4.5) and their estimated parameters, we may compute $\Delta V, \Delta t, \beta$. After that, α is obtained from the LL CDF.

The part $(t > t_{min})$ of EQUATION (4.3) may be rewritten as:

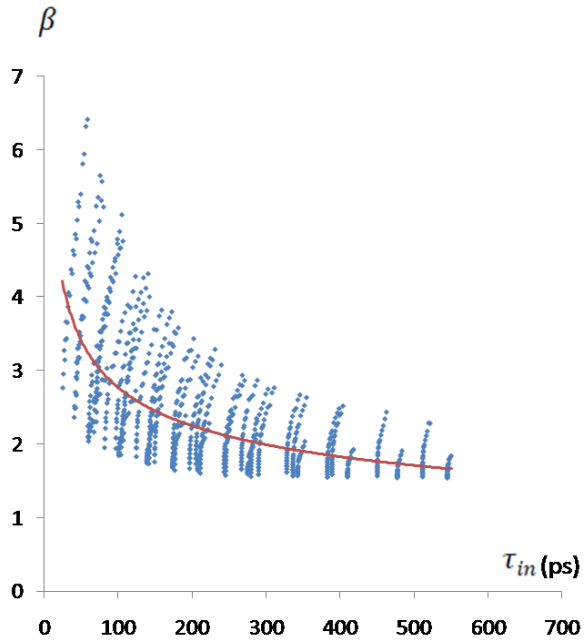
$$t = t_{min} + \tau_{in} \cdot \alpha \cdot \left(\frac{V_{dd} - \hat{V}}{\hat{V} + \Delta V}\right)^{-\frac{1}{\beta}} \qquad (t > t_{min}) \tag{4.6}$$



(a) $\Delta V = \frac{0.45}{15.52+1.81\times10^{11}\cdot\tau_{in}}$



(b) $\Delta t = 4.7 \times 10^{-11} + 0.04 \cdot \tau_{in}$



(c) $\beta = \frac{1.32}{0.33+6.9\times10^{9}\cdot\tau_{in}} + 1.51$

FIGURE 4.4 Proposed simple functions

Replacing $t, \hat{V}$ in EQUATION (4.6) with the two points $(t_1, 0.2 \cdot V_{dd})$ and $(t_2, 0.8 \cdot V_{dd})$, we get:

$$t_2 - t_1 = \tau_{in} \cdot \alpha \cdot \left[\left(\frac{0.2}{0.8 + \frac{\Delta V}{V_{dd}}} \right)^{-\frac{1}{\beta}} - \left(\frac{0.8}{0.2 + \frac{\Delta V}{V_{dd}}} \right)^{-\frac{1}{\beta}} \right] = 0.6 \cdot \tau_{in} \tag{4.7}$$

Then, parameter α is computed by:

$$\alpha = 0.6 \cdot \left[\left(\frac{0.2}{0.8 + \frac{\Delta V}{V_{dd}}} \right)^{-\frac{1}{\beta}} - \left(\frac{0.8}{0.2 + \frac{\Delta V}{V_{dd}}} \right)^{-\frac{1}{\beta}} \right]^{-1} \tag{4.8}$$

Because of the dispersion of the points about the estimated LSM functions, the above way to identify parameters $\alpha, \beta, \Delta t, \Delta V$ leads to loss of accuracy. However, among the information about a signal, only τ_{in} is available during the procedure of computation in FIGURE 3.4. Besides, such a way is simple to apply under HSPICE [43], which is not good at handling complex mathematical functions.

Next, we compare the accuracy of linear ramp and LL-based model following the procedure below:

a) Collect 500 output signals under various conditions, like cell type, temperature, etc.;
b) For each signal, measure the 20% – 80% slope and approximate the signal respectively by linear ramp and LL-based model;
c) Normalize all measured and approximated signals by its own slope, and transform the point $(t_0, 0.5 \cdot V_{dd})$ of each signal to point $(0, 0.5 \cdot V_{dd})$, as shown in FIGURE 4.5;
d) Define a series of points, e.g. $-1.5, -1.4, \ldots, 0, \ldots, 1.4, 1.5$, and compute errors point by point for each linear ramp and LL-based approximation;
e) Compute the average errors of signal samples: all linear ramp approximations, LL-based approximations with slopes respectively less than 100 ps and 200 ps, and all LL-based approximations, as shown in FIGURE 4.6.

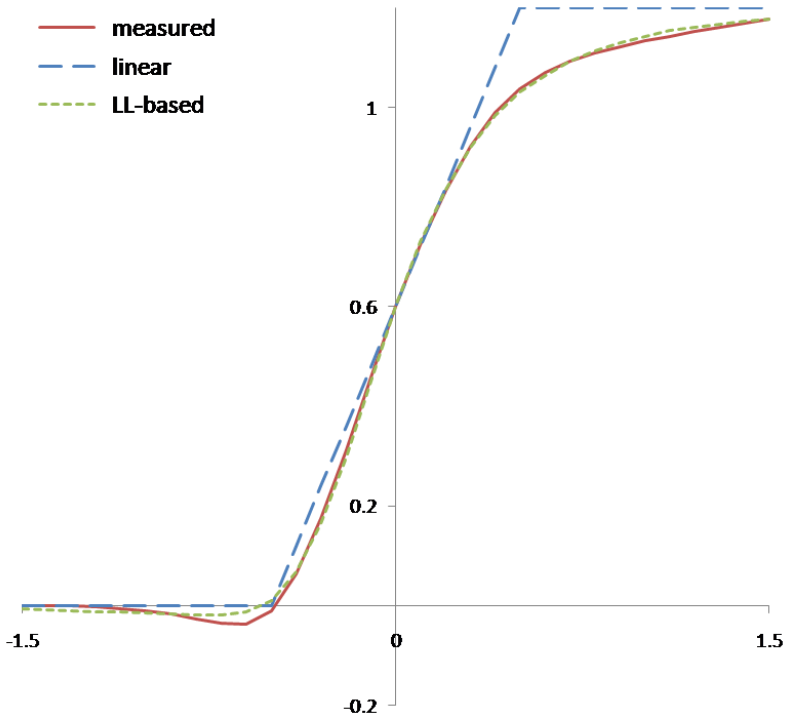


FIGURE 4.5 Normalized and transformed signals

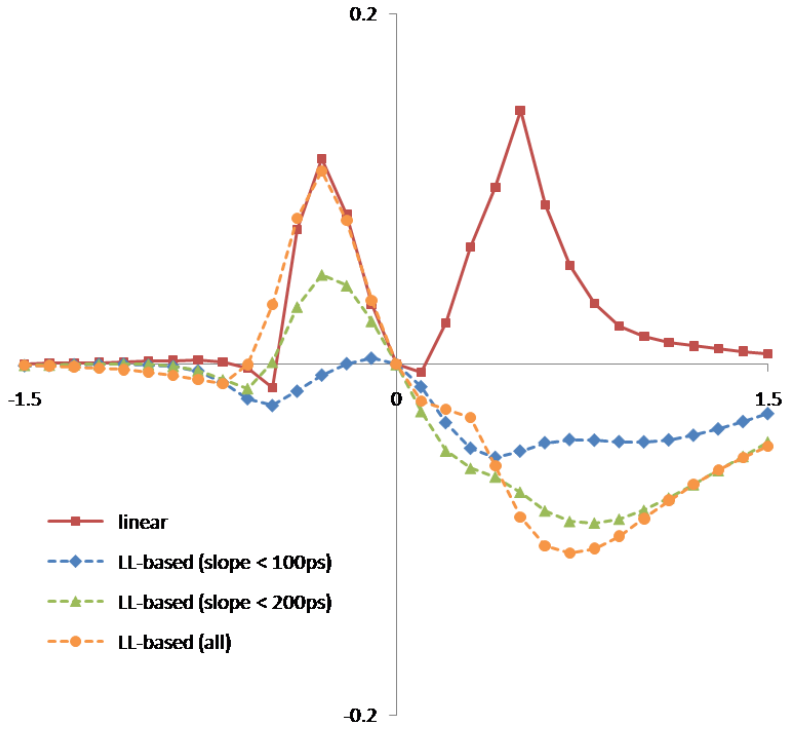


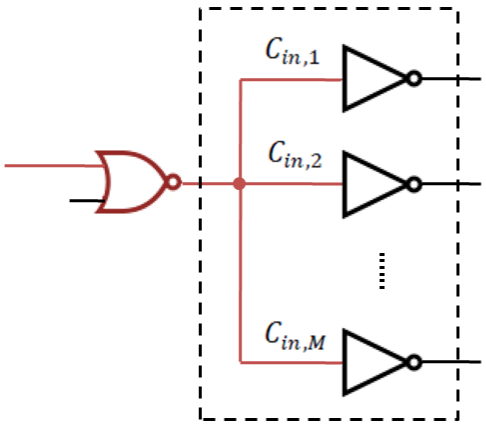
FIGURE 4.6 Average errors of approximated signals (65 nm)

In FIGURE 4.6, linear ramp model has positive errors on both sides of the vertical axis, while LL-based model has negative errors on the right side. In addition, LL-based approximations are better for small slopes than large slopes. In this figure, it is not clear whether LL-based model is more accurate or not. However, latter comparisons in SECTION 4.1.3 show that LL-based model is better in capturing slope variations during characterization.

4.1.2 Output load variations

The main drawback of modeling output load with capacitors during timing characterization, as in FIGURE 4.1(a), is that they are not able to capture output load variations, or more precisely, the impact of load variations on cell timings. This weakness is due to the fact that the charge of a capacitor keeps constant, whereas in real circuit, the charges of input pins of all cells depend on process parameters, and therefore are random variables.

In order to better represent what happens around a cell in real circuits, our timing characterization uses inverters instead of capacitors to model output load. As shown in FIGURE 4.7, we connect M inverters at the output pin of the considered cell. The sum of input charges of all inverters is considered as the nominal value of output load. Note that these inverters can be of different input charges. As mentioned before, input charges of cells, inverters included, depend on process parameters, which are random during MC simulations. Thus, the model using inverters captures output load variations during characterization. In others words, output load variations are contained in conditional variances of timing variables.



$$C_{out} = \sum_{m=1}^{M} C_{in,m}$$

FIGURE 4.7 M inverters as output load

4.1.3 Comparison

The validation of the SSTA engine given in TABLES 3.4 – 3.5 is done with a statistical timing library constructed by data that is collected with the LL-based signal model and inverters as output load. In other words, these tables validate the proposed signal model and the use of inverters as output load model. In order to demonstrate the effects of using these models, we compare standard deviations of path delays estimated by data from MC simulations, which are considered as reference values, with those computed using statistical timing libraries based on the following combinations of input signal and output load models:

- LL-based signal and inverters,
- linear signal and inverters,
- LL-based signal and capacitors.

TABLE 4.1 gives some examples. Comparing the average relative errors of these combinations, we find a difference in percentage of about 16 between the first two combinations, and about 12 between combinations 1 and 3. In addition, results of the last two columns are all underestimated, which are unexpected in case of statistical timing analysis. This table gives the conclusion that both LL-based input signal and inverters output load improve the accuracy of computing path delay standard deviations.

TABLE 4.1 Comparisons of path delay standard deviations computed with statistical timing libraries based on different combinations of input signal and output load models (65 nm)

path	logical depth	σ_{pd} (ps)	$\frac{\hat{\sigma}_{pd} - \sigma_{pd}}{\sigma_{pd}}$ %		
			LL-based signal + inverters (combination 1)	linear signal + inverters (combination 2)	LL-based signal + capacitors (combination 3)
1	5	22.9	7.0%	−6.8%	−4.6%
2	10	36.2	7.7%	−4.5%	−4.8%
3	15	45.4	0.9%	−14.6%	−10.7%
4	20	57.7	6.4%	−11.1%	−5.7%
5	25	57.8	2.8%	−17.2%	−7.2%
average			5.0%	−10.84%	−6.6%

4.1.4 Weaknesses

The weakness of LL-based signal model can be found in FIGURE 4.2. The derivatives $\frac{dH(t)}{dt}$ of signals from circuit simulations converge rapidly to 0 after the corresponding maximum point; whereas the PDFs of LL distributions seem to converge to 0 not as fast as the derivatives of signals do. This problem is also shown in FIGURE 4.6, where LL-based model has negative errors on the right side of the vertical axis. Thus, this weakness of signal model would be a point to address for higher accuracy of approximation.

As regards the output load model, its weakness is obvious, because we have no argument to support:

a) the use of inverters instead of other cells,
b) the structure of inverters which have different charges at input pin,
c) the number of inverters connected at the output pin of the considered cell.

4.2 ACCELERATION TECHNIQUES

Timing characterization is implemented by running MC simulations, which demand very high computational cost. Even though this step of characterization is only a one-time job as stated in CHAPTER 3, it is important to accelerate its procedure to reduce runtime. Such a goal can be achieved by reducing either the number of runs, or the number of points to characterize, or both.

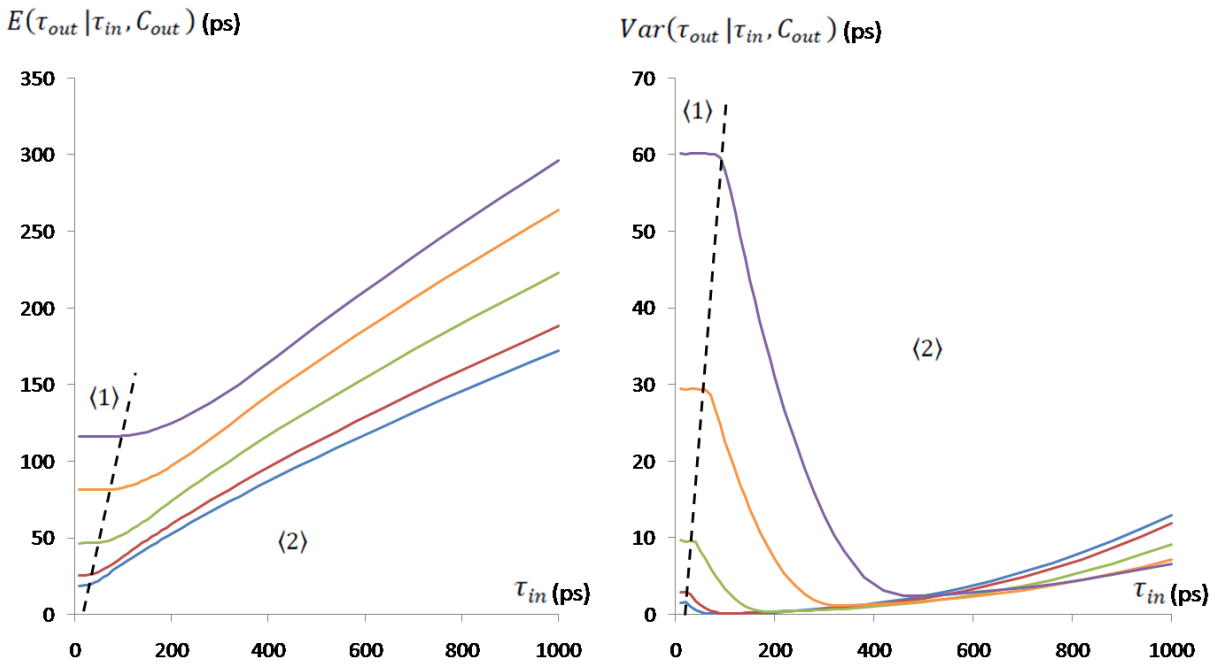
The first way will lead to a loss of accuracy when estimating conditional moments if we continue using classic MC techniques. Theoretically, we may use variants, like importance sampling to reduce the number of runs without losing too much accuracy in estimating conditional moments. However, applying this variant sampling technique on dozens of process parameters is complicated and its accuracy is not clear.

The second way will worsen the accuracy of the approximating function shown in FIGURE 3.3, if it leads to a reduction of the number of points contained in each lookup table. This conclusion is because when approximating a nonlinear function with linear interpolation, the accuracy will

be worse with fewer interpolating points. Note that the number of points that need to be characterized is not exactly the same as the number of points of each lookup table. The technique of acceleration presented below reduces the number of points to characterize while keeping the same number of points in each lookup table.

4.2.1 Reducing dimension

In FIGURE 4.8(a), the curves show how the conditional output slope mean of an inverter $E(\tau_{out}|\tau_{in}, C_{out})$ varies along with τ_{in}. Each of these curves corresponds to a value of the output load C_{out}. They are constant in region ⟨1⟩, called ***Fast Input Range*** (FIR), while in region ⟨2⟩, called ***Non-Fast Input Range*** (N-FIR), $E(\tau_{out}|\tau_{in}, C_{out})$ varies. FIGURE 4.8(b) shows that there exists FIR and N-FIR as well for the conditional output slope variance $Var(\tau_{out}|\tau_{in}, C_{out})$. Note that the FIRs for $E(\tau_{out}|\tau_{in}, C_{out})$ and $Var(\tau_{out}|\tau_{in}, C_{out})$ are identical for a given value of output load.



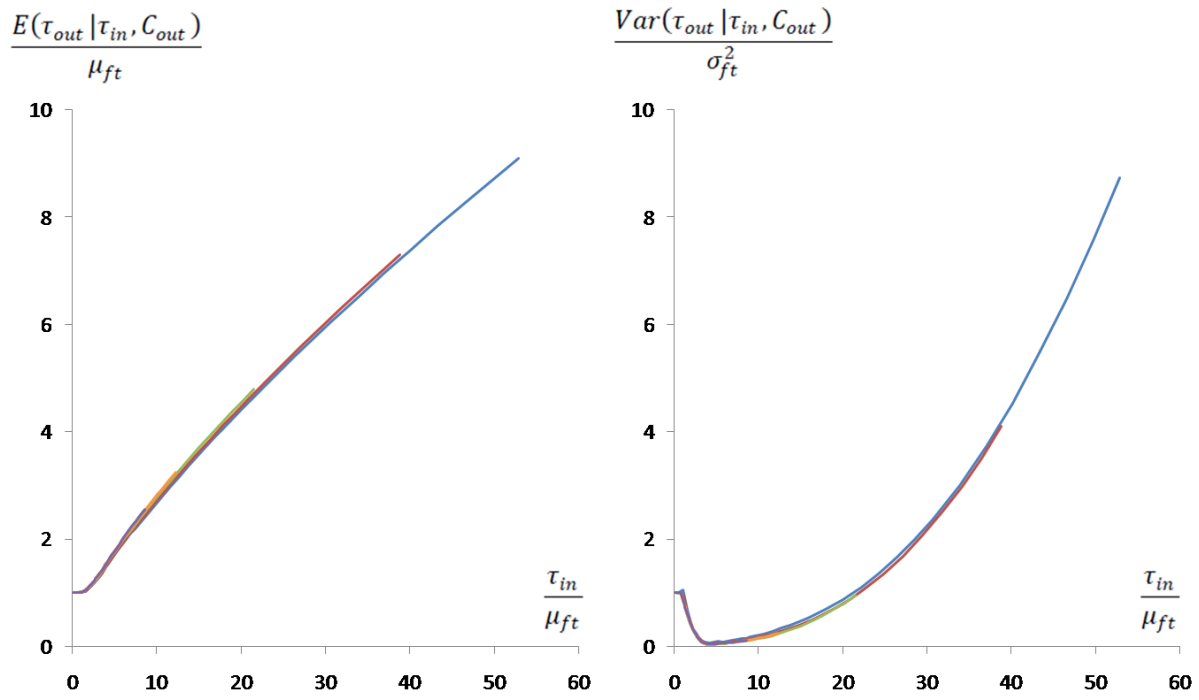
(a) Conditional means of output slopes

(b) Conditional variances of output slopes

FIGURE 4.8 Illustrations of FIR and N-FIR

Given a value c of C_{out}, we define τ_{th}^{c} the threshold between FIR and N-FIR corresponding to c. If $\tau_{in} \in (0, \tau_{th}^{c})$, then $E(\tau_{out}|\tau_{in}, C_{out} = c)$ and $Var(\tau_{out}|\tau_{in}, C_{out} = c)$ are constants, which we denote respectively as $\mu_{\tau_{out}^{ft}}(c)$ and $\left[\sigma_{\tau_{out}^{ft}}(c)\right]^2$, or as μ_{ft} and σ_{ft}^2 for simplicity.

Next, we divide the axes $E(\tau_{out}|\tau_{in}, C_{out})$ and τ_{in} in FIGURE 4.8(a) by μ_{ft}. This normalization, shown in FIGURE 4.9(a), transforms all the curves in FIGURE 4.8(a) into a unique one (up to slight discrepancies) that we call the ***standard curve***. This standard curve is independent of C_{out}. Similarly, normalizing in the same way $Var(\tau_{out}|\tau_{in}, C_{out})$ and τ_{in} by respectively σ_{ft}^2 and μ_{ft} also produces a standard curve for $Var(\tau_{out}|\tau_{in}, C_{out})$.

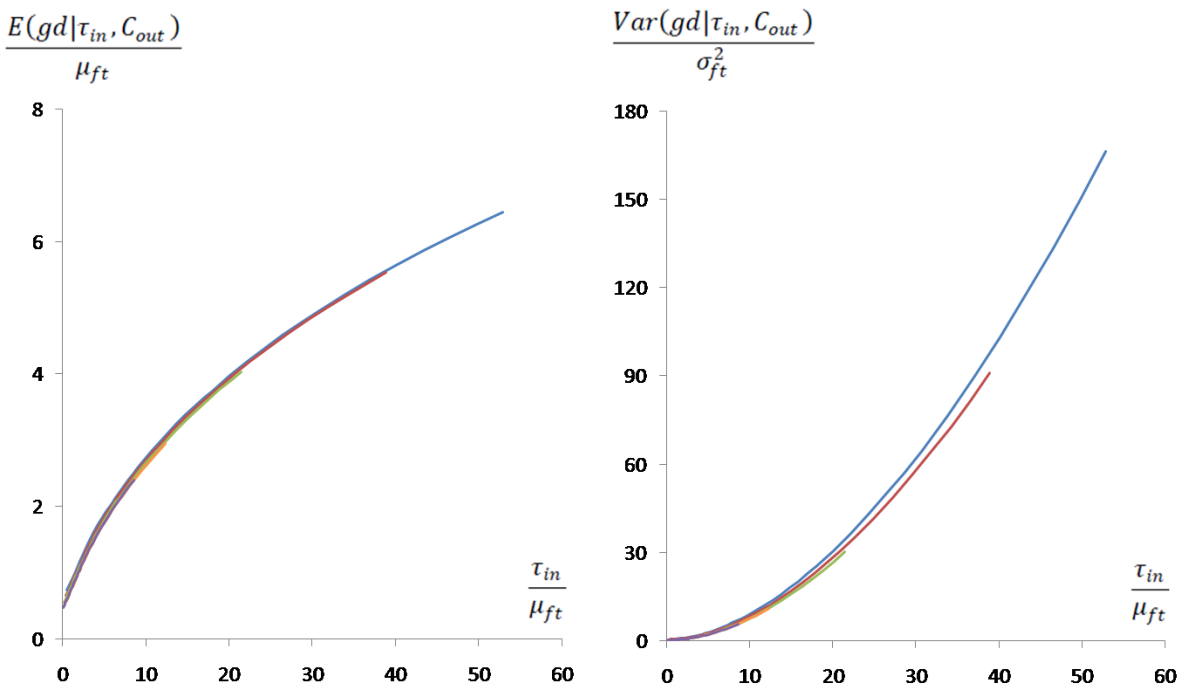


(a) Standard curve for $E(\tau_{out}|\tau_{in}, C_{out})$

(b) Standard curve for $Var(\tau_{out}|\tau_{in}, C_{out})$

FIGURE 4.9 Illustration of normalized conditional moments of output slope

As shown in FIGURE 3.2, output slope τ_{out} and cell delay gd are the two considered timing variables. If we do the same normalization to the conditional delay moments of the same type of cell as above, we find as well standard curves independent of C_{out}, as illustrated in FIGURE 4.10.



(a) Standard curve for $E(gd|\tau_{in},C_{out})$ (b) Standard curve for $Var(gd|\tau_{in},C_{out})$

FIGURE 4.10 Illustration of normalized conditional moments of cell delay

Hence, functions of conditional moments depending on the two factors τ_{in} and C_{out}, can be transformed into functions of only one factor $\frac{\tau_{in}}{\mu_{ft}}$ as follows:

$$\begin{cases} \dfrac{E(\tau_{out}|\tau_{in},C_{out})}{\mu_{ft}} = h_1\left(\dfrac{\tau_{in}}{\mu_{ft}}\right) \\ \dfrac{Var(\tau_{out}|\tau_{in},C_{out})}{\sigma_{ft}^2} = h_2\left(\dfrac{\tau_{in}}{\mu_{ft}}\right) \\ \dfrac{E(gd|\tau_{in},C_{out})}{\mu_{ft}} = h_3\left(\dfrac{\tau_{in}}{\mu_{ft}}\right) \\ \dfrac{Var(gd|\tau_{in},C_{out})}{\sigma_{ft}^2} = h_4\left(\dfrac{\tau_{in}}{\mu_{ft}}\right) \end{cases} \tag{4.9}$$

where μ_{ft} and σ_{ft}^2 are identified, according to [50], by:

$$\begin{cases} \mu_{ft} = A + B \cdot C_{out} \\ \sigma_{ft}^2 = B^2 \cdot C_{out}^2 \end{cases} \tag{4.10}$$

Here, a small number of simulations yields to an accurate estimate of A, B and this requires very little runtime.

4.2.2 Discussion

In CHAPTER 3, the input slope and output load indices of each lookup table have respectively 9 and 6 values. Therefore, a table has 54 points to characterize. However, with the EQUATIONS (4.9) – (4.10), only ten points or so need to be characterized. In FIGURE 4.11, we plot all the normalized points contained in a table of conditional output slope mean. This figure shows that this procedure leads, with 10 points, to approximately the same accuracy as that of a 54 points table.

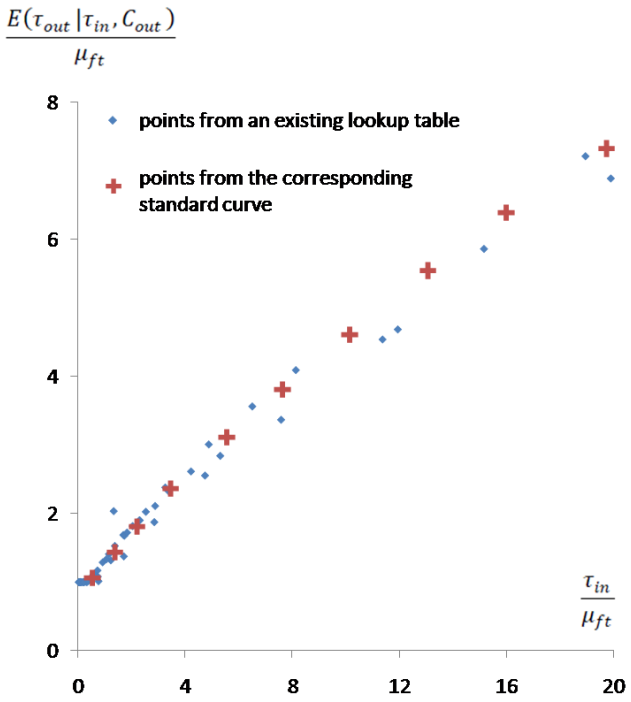
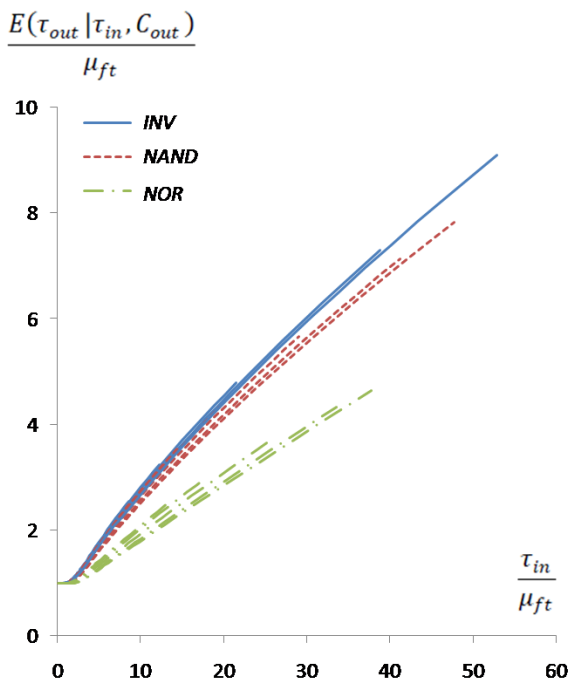
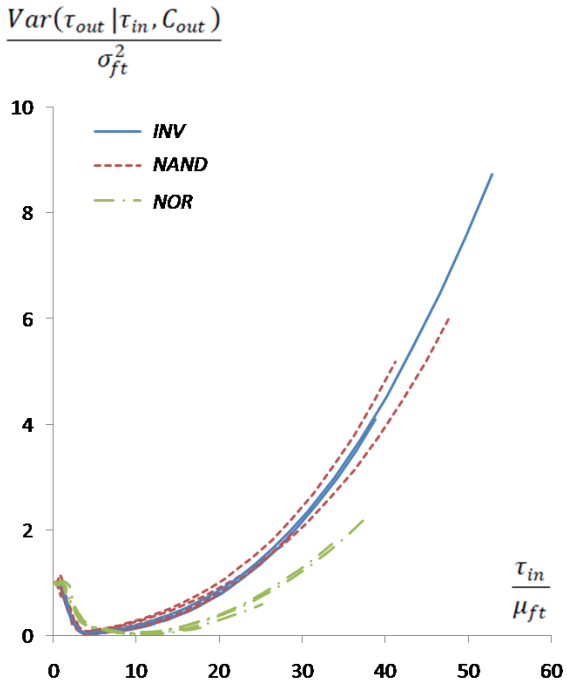


FIGURE 4.11 Reduction of points to characterize

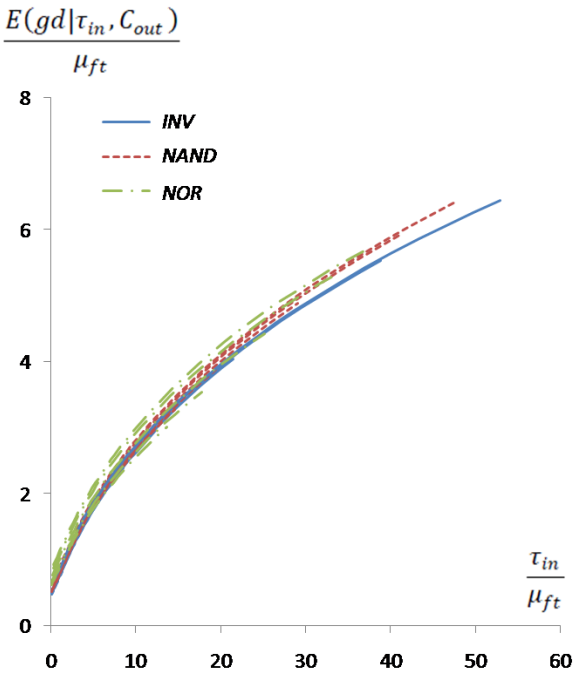
Although this technique accelerates the procedure of timing characterization, its accuracy should be carefully studied before application. To illustrate this, FIGURE 4.12 compares the normalized curves of a NOR, $NAND$, INV cell respectively for cases $E(\tau_{out}|\tau_{in}, C_{out})$, $Var(\tau_{out}|\tau_{in}, C_{out})$, $E(gd|\tau_{in}, C_{out})$ and $Var(gd|\tau_{in}, C_{out})$.



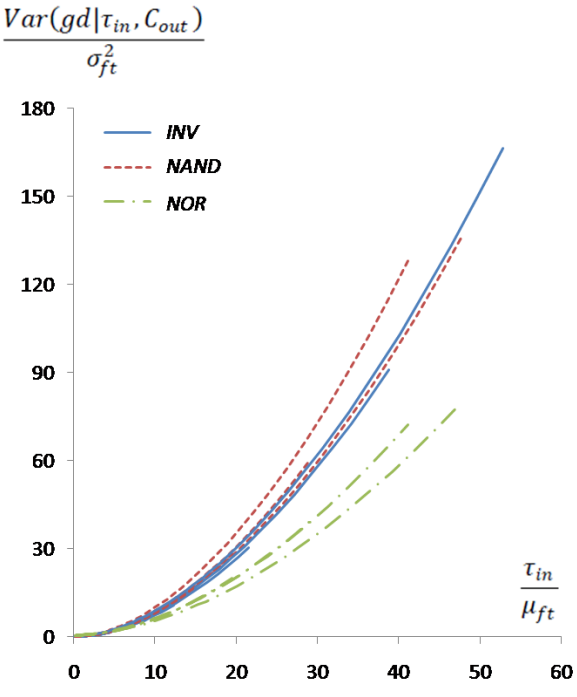
(a) Normalized curves for case of $E(\tau_{out}|\tau_{in},C_{out})$



(b) Normalized curves for case of $Var(\tau_{out}|\tau_{in},C_{out})$



(c) Normalized curves for case of $E(gd|\tau_{in},C_{out})$



(d) Normalized curves for case of $Var(gd|\tau_{in},C_{out})$

FIGURE 4.12 Comparisons of normalized curves

From this figure, if we consider one of the normalized curves as standard curve for the $NAND$ and NOR cell, then their accuracies will not be as good as for an inverter. In consequence, to profit from this acceleration technique, we should study its accuracy for more cells or find out a way to identify standard curves which provide acceptable results.

4.3 SUMMARY

Instead of the conventional method, we use the LL distributions to approximate input signals and inverters to model output load during timing characterization. These improvements allow us to better capture slope and load variations. In addition, to save CPU time of characterization, the reducing dimension technique is proposed. However, more work should be done to apply this promising technique into practice.

CHAPTER

5

COMPARISONS AND APPLICATIONS

In this chapter, we put the SSTA framework into practice and compare its results with those of CTA. SECTION 5.1 gives some examples to show the gain of our SSTA engine. In SECTION 5.2, we talk about ordering of critical paths, i.e. order of paths in terms of decreasing delays. The discrepancy between orderings obtained respectively by SSTA and CTA is explained. In SECTION 5.3, we study the factors that affect cell-to cell delay correlation. This is a first step toward the goal of optimizing circuit design with delay correlations.

As the next generation of timing tool, ***Statistical Static Timing Analysis*** (SSTA) is compared to its predecessor ***Corner-based Timing Analysis*** (CTA) in various aspects, such as accuracy and runtime. An important concern among these aspects is the gain of using SSTA relative to CTA, because this gain, in a sense, declares whether SSTA is a promising replacement. In SECTION 5.1, we focus on the gain of using our SSTA engine.

5.1 GAIN OF SSTA

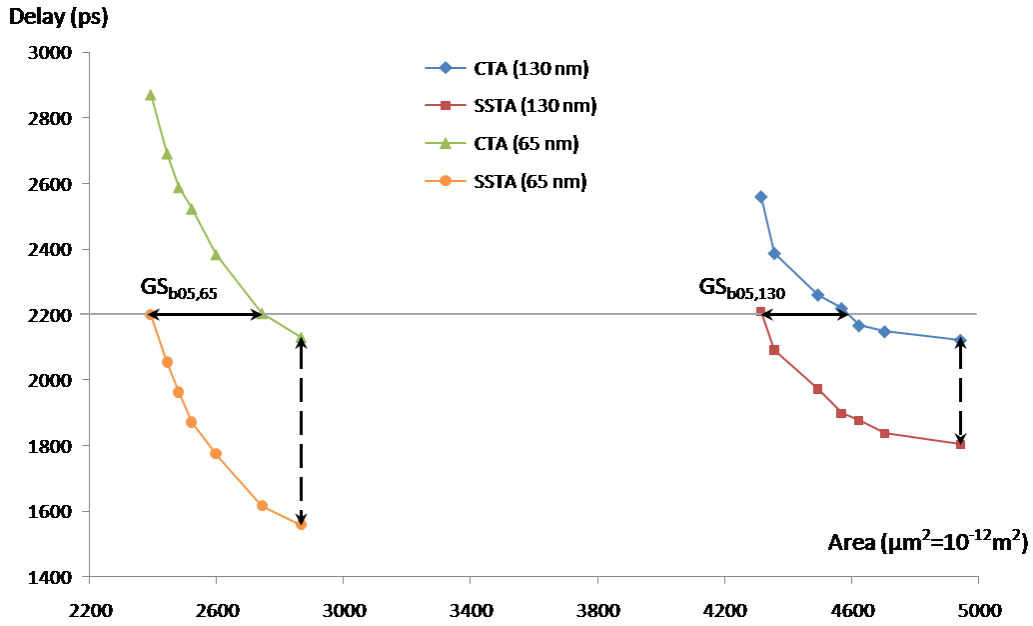
The goal of IC design is to produce a circuit which implements intended functions, occupies minimal area and meets the timing constraints. To be more precise, if a designer is given a number of circuit implementations of the same functionality, he would select the implementation with the minimal area among those meeting a particular delay. This is reasonable because a circuit with smaller area usually consumes less power during operation. Thus, to demonstrate the gain of SSTA, a common way is, for a function to implement and a given delay (or a clock period), to compare area of circuits, which are obtained respectively with CTA and SSTA.

For this comparison, we first construct area-delay curves to circuits b05 and b07 of the ITC99 benchmarks according to the following procedure:

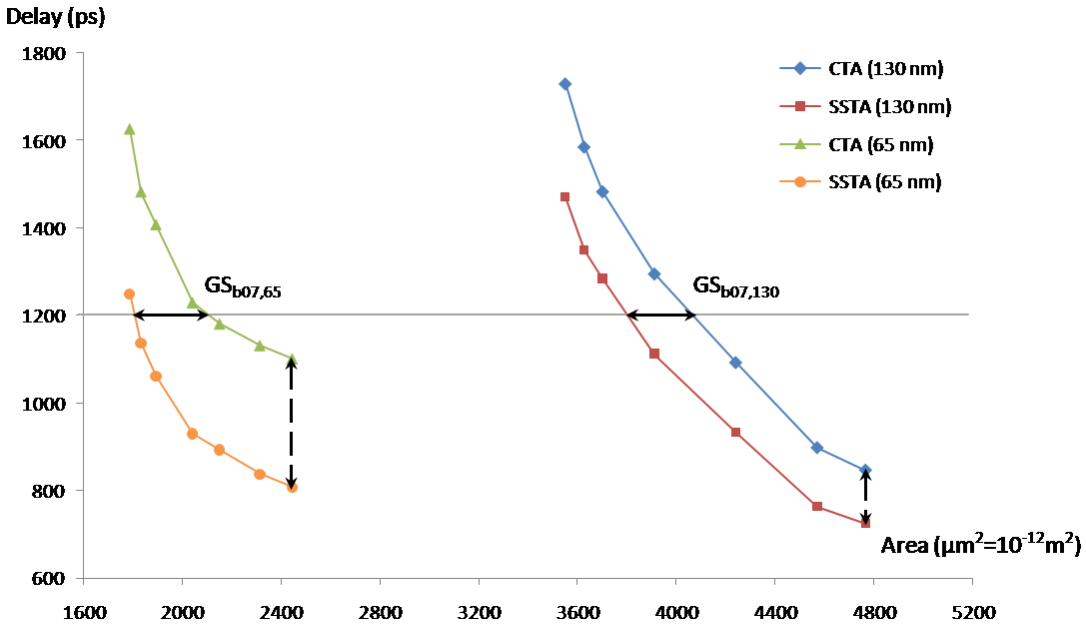
a) Define a series of clock periods T_{CLK_m}, $(m = 1, 2, \dots, 7)$ respectively for b05 and b07;
b) For each clock period T_{CLK_m}, produce an implementation IPN_m with RTL Compiler [45], and note the corresponding circuit area cs_m;
c) For the implementation IPN_m, extract a set $U_{100,m}$ of 100 critical paths in terms of decreasing worst delays w_{pd_u}, $\left(u \in U_{100,m}\right)$ obtained by CTA;
d) Under the worst environmental conditions: $125℃$ (temperature) and $1.1V$ (supply voltage), compute path delay means μ_{pd_u} and variances $\sigma^2_{pd_u}$ for the set $U_{100,m}$ with our SSTA engine;
e) Compute w_{IPN_m} and s_{IPN_m} by:

$$\begin{cases} w_{IPN_m} = \underset{u \in U_{100,m}}{max} \left(w_{pd_u}\right) \\ s_{IPN_m} = \underset{u \in U_{100,m}}{max} \left(\mu_{pd_u} + 3 \cdot \sigma_{pd_u}\right) \end{cases} \tag{5.1}$$

f) Plot the points of CTA (cs_m, w_{IPN_m}) and SSTA(cs_m, s_{IPN_m}) and approximate the area-delay curves by these points and linear interpolation, as shown in FIGURE 5.1.



(a) Area-delay curves of b05



(b) Area-delay curves of b07

FIGURE 5.1 Gains of SSTA for circuits b05 and b07

In FIGURE 5.1(a), the length of the solid double arrow $GS_{b05,130}$ represents the difference of area between the CTA curve and the SSTA curve when circuit delay is set to 2200 ps (as an example) in the case of 130 nm technology. The length of the solid double arrows $GS_{b05,65}$, $GS_{b07,130}$ and $GS_{b07,65}$ have similar meanings. From this figure, it is obvious that, for a given delay, the area of circuits implemented with SSTA is smaller than the area of the corresponding implementations using CTA in both 130 nm and 65 nm cases. Define the percentage of gains in area as:

$$r_{GS} = \frac{GS}{S_{CTA}}\% \tag{5.2}$$

where S_{CTA} is the corresponding area value of CTA. Then, r_{GS} for the delay 2200 ps in FIGURE 5.1(a) and 1200 ps (also set as an example) in FIGURE 5.1(b) are:

$$\begin{cases} r_{GS_{b05,130}} = 5.5\% \\ r_{GS_{b05,65}} = 12.8\% \\ r_{GS_{b07,130}} = 6.1\% \\ r_{GS_{b07,65}} = 13.7\% \end{cases}$$

According to the area-delay curves in FIGURE 5.1, there exist very few values of delay where the horizontal double arrows (i.e. area gains) are bounded. For example, in FIGURE 5.1(a), the SSTA curve of 65 nm gives no value of area at 2600 ps while the CTA curve does. This forbids a proper comparison of area. Thus, for an alternative comparison, we consider the vertical distances between each couple of curves which corresponds to gains of delay (dashed double arrows in FIGURE 5.1) for a given area. These distances can be computed at more area points, which allows considering the two following average gains of delays:

$$\overline{GD} = \frac{1}{7} \cdot \sum_{m=1}^{7} \left(w_{IPN_m} - s_{IPN_m} \right) \tag{5.3}$$

$$\overline{r_{GD}} = \frac{1}{7} \cdot \sum_{m=1}^{7} \frac{w_{IPN_m} - s_{IPN_m}}{w_{IPN_m}}\% \tag{5.4}$$

where w_{IPN_m}, s_{IPN_m} are defined in EQUATION (5.1).

TABLE 5.1 shows the gains defined in EQUATIONS (5.3) – (5.4). According to this table, we can draw the following two conclusions:

a) In terms of circuit, $\overline{GD}$ values of circuit b05 are about two times larger than those of circuit b07 in both 130 nm and 65 nm technology. The two couples of $\overline{r_{GD}}$ values are much closer, e.g. 2.6% vs. 1.0% and 13.6% vs. 12.3%. These comparisons indicate that $\overline{GD}$ increases along with path logical depth whereas the normalized gain $\overline{r_{GD}}$ does not depend on circuit.
b) In terms of technology, both $\overline{GD}$ and $\overline{r_{GD}}$ of the two circuits in the 65 nm technology are much larger than those in the 130 nm technology. It is predicted that these two average gains of delays will become more and more important as the feature size shrinks from 65 nm to 45 nm, 32 nm, etc.

TABLE 5.1 Average delay gains of the SSTA engine over CTA (without interconnects)

name	technology	maximal path depth	$\overline{GD}$ (ps)	$\overline{r_{GD}}$ (%)
b05	130 nm	18	50	2.6%
	65 nm	27	295	13.6%
b07	130 nm	12	11	1.0%
	65 nm	17	140	12.3%

5.2 ORDERING OF CRITICAL PATHS

For the time being, most IC designers still use tools based on CTA, and consider SSTA as a complement to CTA, which may lead to cases where results of SSTA and those of CTA do not coincide. In this section, we show and explain the discrepancy between orderings obtained respectively by SSTA and CTA.

A typical ***Computer-Aided-Design*** (CAD) tool based on CTA, like RTL Compiler [45], may extract a set of N critical paths for optimization of circuit design. These N critical paths are ordered by decreasing worst delay according to CTA, i.e. the first critical path has the maximal worst delay; the second one has the second maximal value, and etc. However, if path delays of

the same set are computed by SSTA, the ordering of these paths may be different with that of CTA.

To illustrate the differences of orderings, we follow the procedure below for the circuit b07 in the 65 nm technology:

a) Choose two valid clock periods, for example: T_{CLK} =1400, 2000 ps;
b) For each clock period, produce an implementation and extract a set U_{100} of 100 critical paths;
c) For each critical path $u_i \in U_{100}$, compute the worst delay $w_{pd_{u_i}}$ by CTA and the corresponding delay under the 125℃ (temperature) and $1.1V$ (supply voltage) operating condition by SSTA:

$$s_{pd_{u_i}} = \mu_{pd_{u_i}} + 3 \cdot \sigma_{pd_{u_i}} \tag{5.5}$$

d) For each critical path $u_i \in U_{100}$, compute the path rank according to worst delay by CTA as:

$$rk_{u_i} = \sum_{j=1}^{100} if\left\{w_{pd_{u_j}} \geq w_{pd_{u_i}}, then\ 1, else\ 0\right\} \tag{5.6}$$

To break ties ($rk_{u_{i_1}} = rk_{u_{i_2}} = \cdots = rk_{u_{i_k}} = M$, for $i_1 < i_2 < \cdots < i_k$), we use the following rule:

$$\begin{cases} rk_{u_{i_1}} = M - k + 1 \\ rk_{u_{i_2}} = M - k + 2 \\ \quad \cdots\cdots \\ rk_{u_{i_k}} = M \end{cases} \tag{5.7}$$

e) Plot the CTA points $\left(rk_{u_i}, w_{pd_{u_i}}\right)$ and those of SSTA $\left(rk_{u_i}, s_{pd_{u_i}}\right)$ with $i = 1, \ldots, 100$, as shown in FIGURE 5.2;
f) Plot the normalized points $\left(rk_{u_i}, \frac{s_{pd_{u_i}}}{w_{pd_{u_i}}}\right)$ with $i = 1, \ldots, 100$, as shown in FIGURE 5.3.

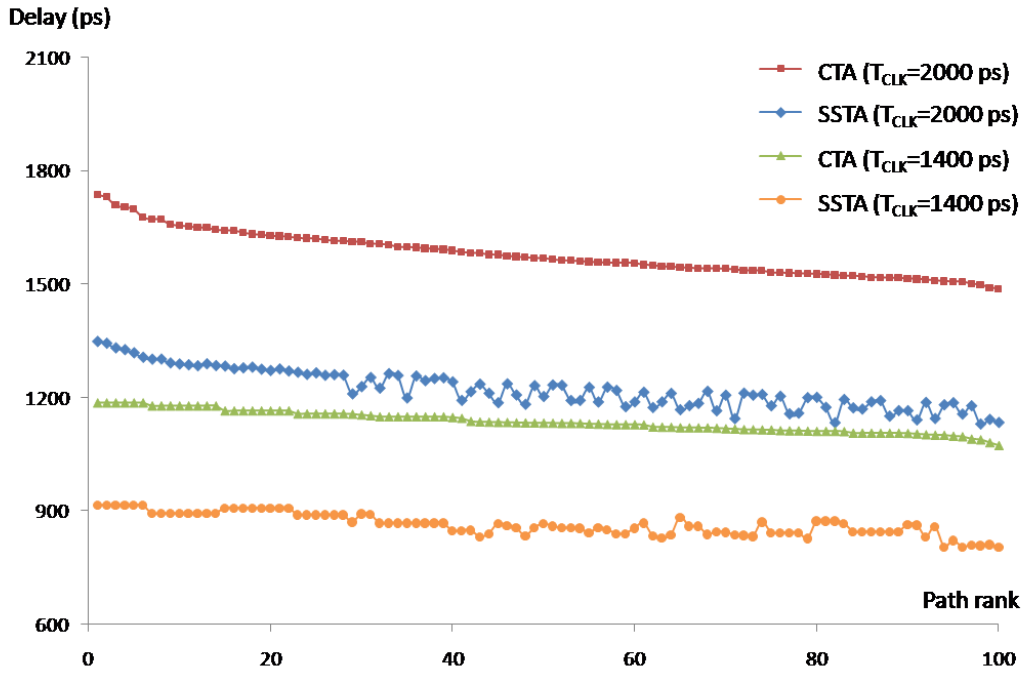


FIGURE 5.2 Delays of ordered critical paths (b07, 65 nm)

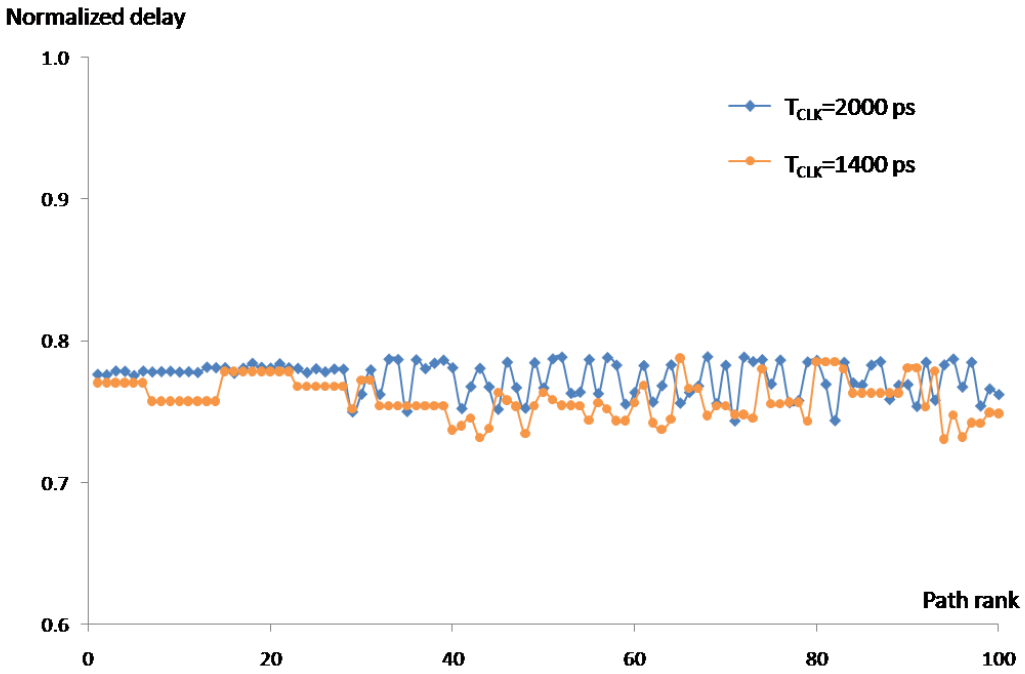


FIGURE 5.3 Normalized delays of ordered critical paths (b07, 65 nm)

According to the above procedure, all paths in FIGURES 5.2 – 5.3 are ordered by decreasing worst delays from CTA. Note the extreme pessimism of CTA with respect to SSTA in FIGURE 5.3, CTA overestimating by 20% to 30% the delay values. Also remark that delays obtained by SSTA are ordered differently from CTA. Indeed, in FIGURE 5.2, comparing the fifty most critical paths provided by CTA and SSTA respectively for cases T_{CLK} =1400 ps and 2000 ps, we find out 42 (respectively 40) common paths among them.

To explain this difference of orderings, we consider a timing path of *K* cells. For each cell, we compute worst cell delay w_{gd_k} by CTA, and cell delay mean μ_{gd_k}, variance $\sigma^2_{gd_k}$ with the SSTA engine under the worst environmental condition: 125℃ and $1.1V$. We can always write w_{gd_k} in terms of μ_{gd_k}, $\sigma^2_{gd_k}$ as a function of θ_k in the following way:

$$w_{gd_k} = \mu_{gd_k} + \theta_k \cdot \sigma_{gd_k} \qquad (k = 1, 2, \dots, K) \tag{5.8}$$

Then, according to EQUATIONS (3.21) and (5.8), the statistical 3σ corner of path delay s_{pd} and the worst path delay w_{pd} can be decomposed as:

$$s_{pd} = \mu_{pd} + 3 \cdot \sigma_{pd} = \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} \rho_{km} \cdot \sigma_{gd_k}\sigma_{gd_m}} \tag{5.9}$$

$$\begin{aligned} w_{pd} &= \sum_{k=1}^{K} w_{gd_k} = \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \left(\sum_{k=1}^{K} \frac{\theta_k \cdot \sigma_{gd_k}}{3}\right) \\ &= \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} 1 \cdot \left(\frac{\theta_k \cdot \sigma_{gd_k}}{3}\right)\left(\frac{\theta_m \cdot \sigma_{gd_m}}{3}\right)} \qquad (5.10a) \\ &= \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} 1 \cdot \left(\frac{w_{gd_k} - \mu_{gd_k}}{3}\right)\left(\frac{w_{gd_k} - \mu_{gd_k}}{3}\right)} \qquad (5.10b) \end{aligned}$$

Comparing EQUATION (5.9) with (5.10a), the constant "1" in EQUATION (5.10a) is replaced by the quantity ρ_{km}. Similarly, $\sigma_{gd_k} \cdot \theta_k/3$ changes to σ_{gd_k} in EQUATION (5.9). Therefore, the discrepancy of orderings comes from two factors: cell-to-cell delay correlation ρ_{km} and standard

deviation of cell delay σ_{gd_k}.

On one hand, in the context of timing analysis, ρ_{km} varies in the interval [0,1]. It is rare that $\rho_{km} = 1$ if $k \neq m$. However, "1" is set to ρ_{km} in EQUATION (5.10a), which introduces a first difference between w_{pd} and s_{pd}.

On the other hand, according to the traditional CTA presented in SECTION 1.2, w_{gd_k} is computed by setting worst corner to each process parameter p_l. Therefore it is unlikely that θ_k is equal to 3 in EQUATION (5.10a), which is second source of difference in orderings.

To identify which factor has more influence on the difference of orderings, we compute $s_{pd}^{'}$ and $s_{pd}^{''}$ with:

$$s_{pd}^{'} = \mu_{pd} + 3 \cdot \sigma_{pd}^{'} \tag{5.11}$$

$$s_{pd}^{''} = \mu_{pd} + 3 \cdot \sigma_{pd}^{''} \tag{5.12}$$

where

$$\sigma_{pd}^{'} = \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} 1 \cdot \sigma_{gd_k}\sigma_{gd_m}} \tag{5.13}$$

$$\sigma_{pd}^{''} = \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} \rho_{km} \cdot \left(\frac{\theta_k \cdot \sigma_{gd_k}}{3}\right)\left(\frac{\theta_m \cdot \sigma_{gd_m}}{3}\right)} \tag{5.14}$$

Comparing EQUATION (5.9) with (5.13), we find out that $\sigma_{pd}^{'}$ eliminates the influence of ρ_{km}. In the same manner, going from EQUATION (5.14) to (5.9) eliminates the influence of θ_k, from EQUATION (5.10a) to (5.13) eliminates the influence of θ_k, and from EQUATION (5.14) to (5.10a) eliminates the influence of ρ_{km}.

The four curves $w_{pd}, s_{pd}, s_{pd}^{'}, s_{pd}^{''}$ in terms of path rank rk_u as defined in EQUATIONS (5.6) – (5.7) are plotted in FIGURE 5.4. From this figure, we can see that the two curves w_{pd} and $s_{pd}^{''}$ have a similar shape, and the same holds for the other couple of curves s_{pd} and $s_{pd}^{'}$. These similarities lead to the conclusion that the discrepancy of orderings comes mainly from the presence of θ_k.

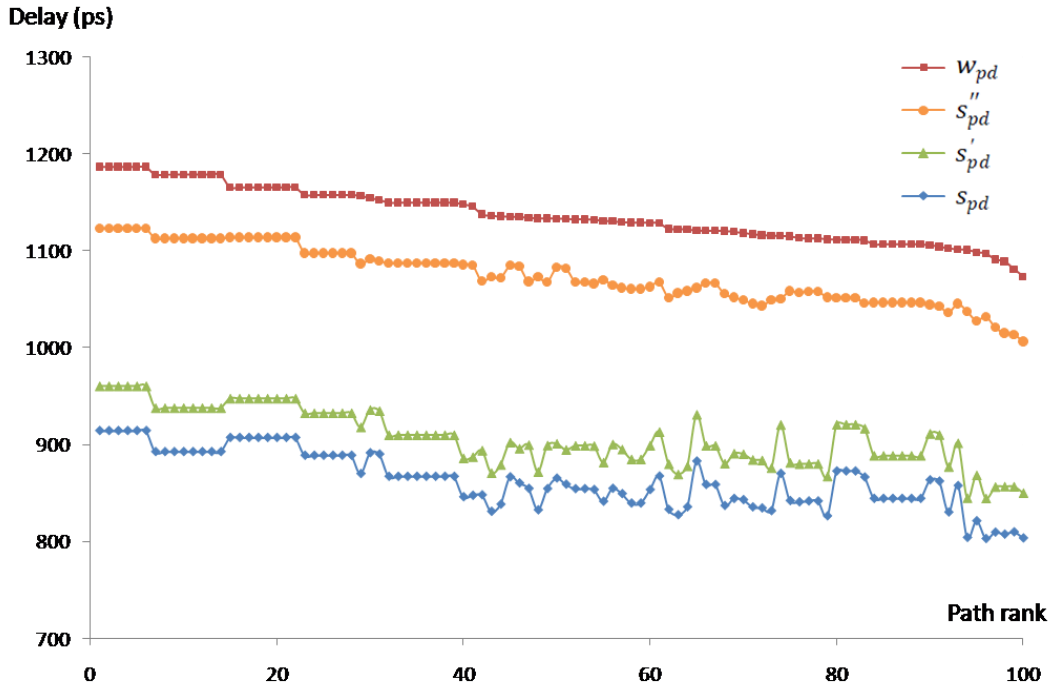


FIGURE 5.4 Interpretation of discrepancy between orderings

Another way of looking at this figure is to consider the gaps between curves. The gaps w_{pd} and $s_{pd}^{'}$ are much larger than those between s_{pd} and $s_{pd}^{'}$. This indicates that the majority of delay gains in using SSTA can be attributed to the gain $\sigma_{gd_k} \cdot (\theta_k/3 - 1)$ of each cell.

5.3 STUDY OF CELL-TO-CELL DELAY CORRELATION

In SECTION 5.2, we highlighted the impact of ***Cell-to-cell Delay Correlation*** (CDC) ρ_{km} on path delay variances. From EQUATION (3.21), reducing CDCs results in smaller path delay

variances and thus constitute, at path level, a main design optimization objective. However, it is not clear whether reducing CDCs meets or not the goal of optimization at circuit level, i.e. to lower the variance defined below:

$$Var(pd_{data} - pd_{clk}) = \sigma_{data}^2 + \sigma_{clk}^2 - 2 \cdot \rho_{dc} \cdot \sigma_{data} \cdot \sigma_{clk} \quad (5.15)$$

where pd_{data}, pd_{clk} are respectively delays of data path and clock path; ρ_{dc} is the ***Path-to-path Delay Correlation*** (PDC) between pd_{data} and pd_{clk}. More details about EQUATION (5.15) were given in SECTION 3.6.1.

As stated before, reducing CDCs gives smaller σ_{data}^2, σ_{clk}^2 AND if at the same time this reduction increases ρ_{dc}, then $Var(pd_{data} - pd_{clk})$ will be smaller. In all other cases, we cannot predict the behavior of this variance and optimization procedure can only be undertaken from case to case, by looking at the values of the elements in EQUATION (5.15). In this section, we take a first step to solve this problem by analyzing how the relative factors influence CDC values.

According to EQUATION (1.1), cell delay is determined by the following two categories of factors:

a) variational factors: process parameters, temperature, supply voltage, input slope and output load

b) fixed factors: cell type, input pin and output edge

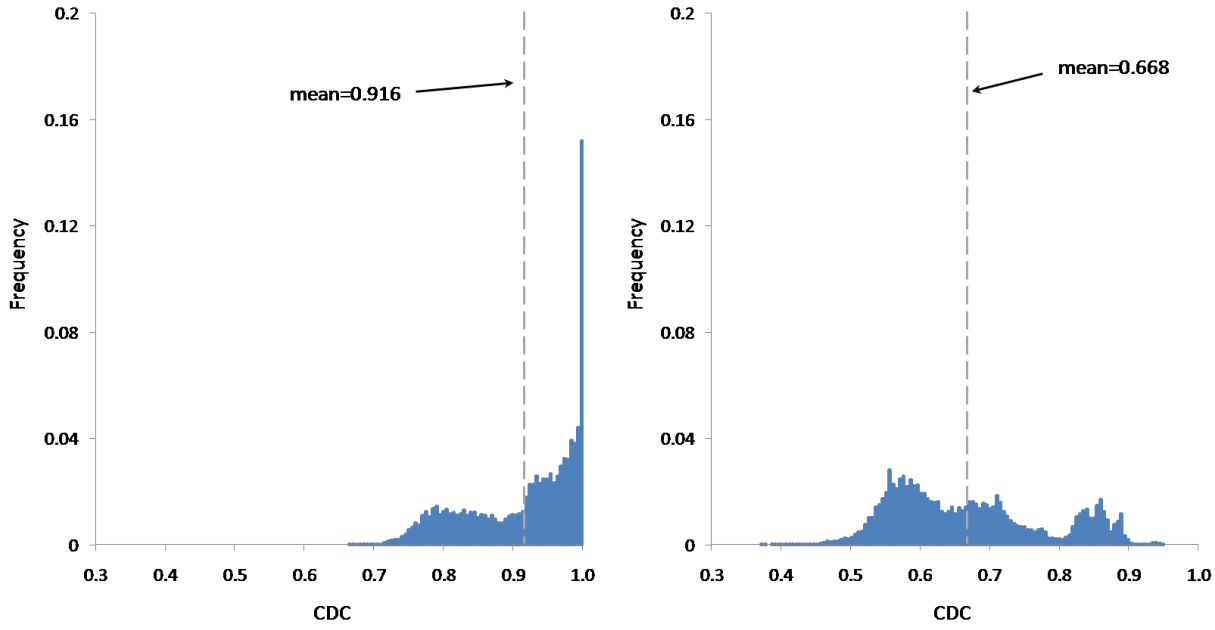
These factors affect CDC as well. However, when computing delays with the SSTA engine, temperature and supply voltage are considered as constants and thus have no influence on CDC. In addition, the relationship between a cell delay and process parameters is not explicitly known. Thus, in SECTION 5.3.1, we focus only on the effect of technology, i.e. the overall effect of process parameters. Two other variational factors: input slope and output load, are studied in SECTION 5.3.2. The effect of fixed factors is the topic of SECTION 5.3.3.

5.3.1 Effect of technology

In TABLE 3.1, the 130 nm and 65 nm technologies are different in number of inter-die and intra-die process parameters. At the same time, process variations of the two technology generations are of great difference. To study the effect of technology, we first extract a total of 3000 paths

from the following circuits: b01, b03, b05, b06 and b07. Then, CDC coefficients of all these paths are computed by the SSTA engine. Finally, from these CDCs, a sample of size 2×10^5 from each technology is drawn randomly for comparison.

FIGURE 5.5 shows the histograms of CDC coefficients. From this figure, CDC coefficients of the 130 nm technology have mean value 0.916 much larger than that of the 65 nm technology which is 0.668. This is explained by the fact that no intra-die process parameter is defined in the 130 nm technology, which results in high CDCs according to EQUATIONS (3.31) and (3.36). In consequence, CDC coefficients of paths implemented in the 65 nm technology are more representative. This technology is preferred in the rest of this section.



(a) 130 nm technology

(b) 65 nm technology

FIGURE 5.5 Histograms of CDC coefficients

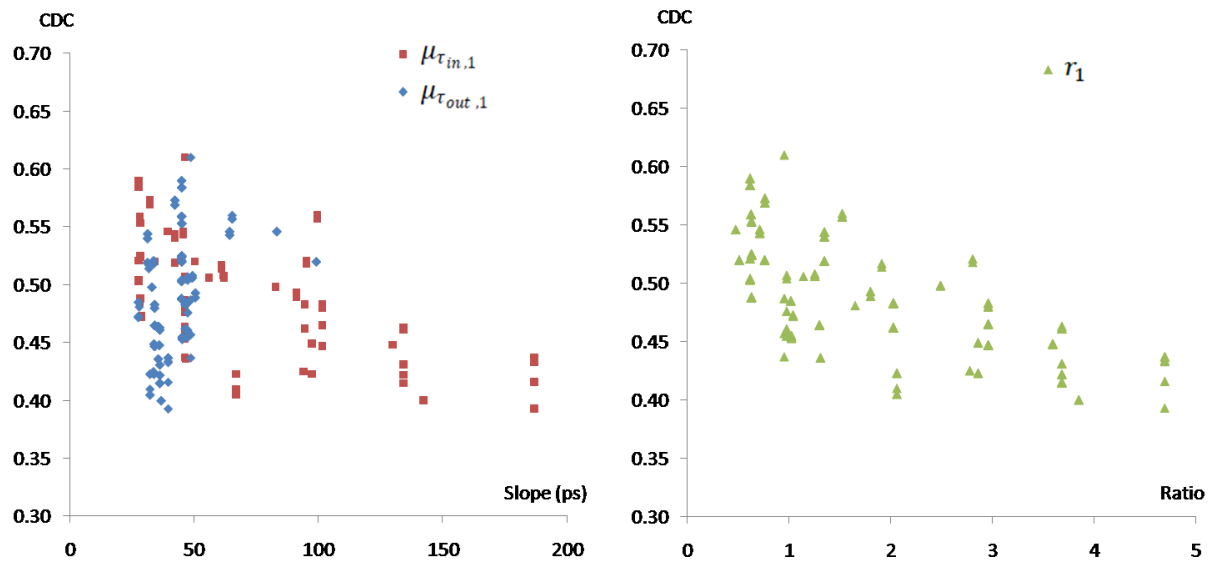
5.3.2 Effect of input slope and output load

As presented in CHAPTER 3, input slope τ_{in} of each cell is considered as a random variable. In this initial work, we only focus on effect of input slope mean $\mu_{\tau_{in}}$. As regards output load, it is replaced by output slope mean $\mu_{\tau_{out}}$ for convenience, which is linear to typical value of output load. Besides, to eliminate effect of fixed factors, among all CDC coefficients, we choose those

with two cells that are of same type, ***input/output*** (I/O) pin and I/O edge. For example, CDC of "$NOR - A/Z - R/F$" indicates CDC between two NOR cells with a rising edge applied at input pin A and a falling edge appearing at output pin Z.

FIGURE 5.6 shows the effects of $\mu_{\tau_{in},1}$, $\mu_{\tau_{out},1}$ and $\frac{\mu_{\tau_{in},1}}{\mu_{\tau_{out},1}}$ (denoted as r_1) on CDC of "$NOR - A/Z - R/F$". In this figure, $\mu_{\tau_{in},1}$ and $\mu_{\tau_{out},1}$ are respectively input slope and output slope mean for the first cell of the couple related to CDC; r_1 does make sense in the context of digital IC. If this ratio is less than a certain threshold, the input slope falls into the ***Fast Input Range*** (FIR) defined in SECTION 4.2.1; if not, it is in the ***Non-Fast Input Range*** (N-FIR). As shown in FIGURE 4.8, output slope means are constants in the FIR; while they vary in the N-FIR. As well, cell delay varies differently in these two ranges, and so does CDC.

In FIGURE 5.6, there exist linear trends of CDC depending respectively on $\mu_{\tau_{in},1}$ and r_1. As the latter term takes into account the effect of $\mu_{\tau_{out},1}$, plus its meaning explained above, we choose it as the considered factor.



(a) Relationship of CDCs and $\mu_{\tau_{in},1}$, $\mu_{\tau_{out},1}$

(b) Relationship of CDCs and r_1

FIGURE 5.6 Effects of $\mu_{\tau_{in},1}$, $\mu_{\tau_{out},1}$ and r_1 on CDCs ($NOR - A/Z - R/F$)

Knowing that correlation is commutative, i.e. $cor(X,Y) = cor(Y,X)$ for any two random variables X and Y, slope factors of the second cell of each couple are added using a commutative

function. Therefore, with three simple commutative functions, we define the following compound ratios:

$$\begin{cases} r_{sum} = r_1 + r_2 \\ r_{dif} = |r_1 - r_2| \\ r_{pdt} = r_1 \cdot r_2 \end{cases} \tag{5.16}$$

where $r_2 = \frac{\mu_{\tau_{in},2}}{\mu_{\tau_{out},2}}$. For the same sample as that in FIGURE 5.6, we compute the compound ratios in EQUATION (5.16) for each CDC coefficient, and plot them in FIGURE 5.7. According to this figure, there is no clear relationship of CDC on r_{dif}; CDC and r_{pdt} seem to have a polynomial relationship; and CDCs seem linear to r_{sum}, which is preferable to describe effect of input and output slope on CDC.

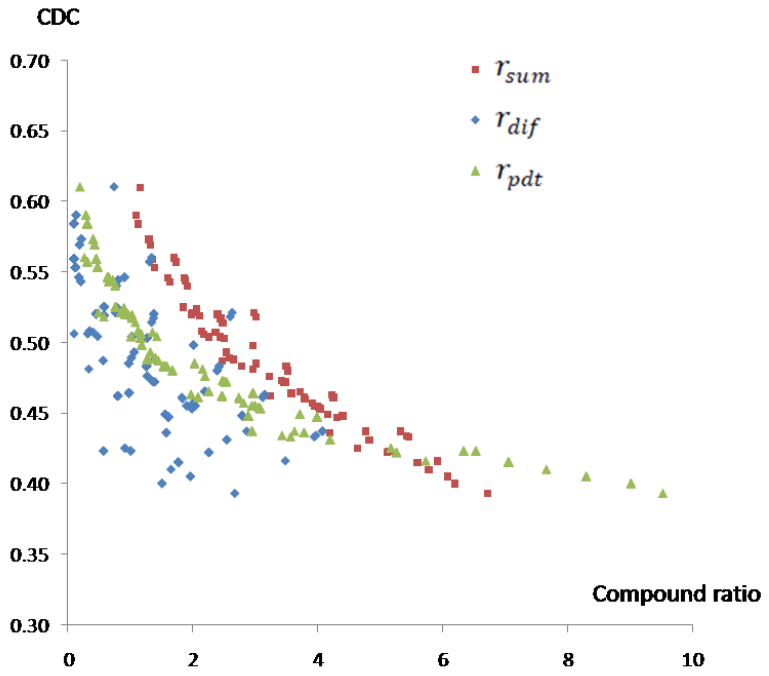


FIGURE 5.7 Relationship of CDCs and different compound ratios ($NOR - A/Z - R/F$)

To confirm the linear dependency of CDC on r_{sum}, we repeat the above procedure for various cell types, I/O pins and I/O edges. FIGURE 5.8 gives four examples. Note that the couple of cells related to the corresponding CDC coefficient have the same cell type, I/O pin and I/O edge. If we

do linear regression for each cloud of points, as shown in the figure, the error of combination "$INV - A/Z - R/F$" is higher than the other three whereas the linear trend is obvious in all cases.

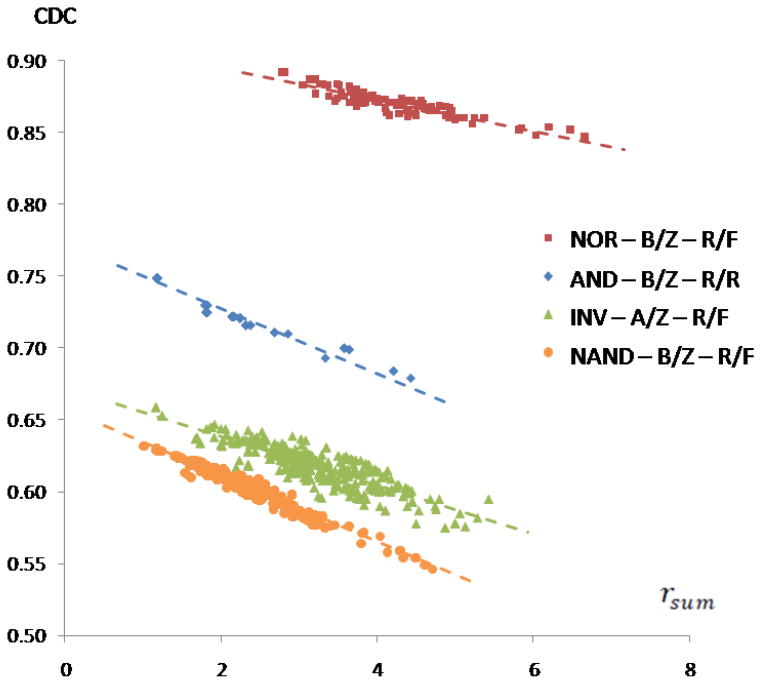


FIGURE 5.8 Relationship of CDCs and r_{sum} for various cell types, I/O pins and I/O edges

5.3.3 Effect of cell type, I/O pin and I/O edge

As described in SECTION 5.3.1, we have a sample of computed CDC coefficients in the 65 nm technology. However, to address the topic of this section, effect of input and output slope should be eliminated by filtering the sample. Typically, we consider the coefficients related to the couple of cells that satisfy the condition below:

$$\begin{cases} r_1 = \dfrac{\mu_{\tau_{in},1}}{\mu_{\tau_{out},1}} \in [0.85, 0.9] \\ r_2 = \dfrac{\mu_{\tau_{in},2}}{\mu_{\tau_{out},2}} \in [0.85, 0.9] \end{cases} \tag{5.17}$$

where $\mu_{\tau_{in},1}, \mu_{\tau_{out},1}, \mu_{\tau_{in},2}, \mu_{\tau_{out},2}$ are respectively slope means of the first and the second cell of the corresponding couple. The interval $[0.85, 0.9]$ is selected for the following reasons:

- If we use the condition $r_1 = r_2 = 0.85$, then the size of the sample after the filtration will be too small for study.
- For any CDC coefficient of the filtered sample, its corresponding compound ratio r_{sum} defined in EQUATION (5.16) is a value the interval $[1.7, 1.8]$, i.e. the difference of any two compound ratios is 0.1, which is acceptable relative to the range of r_{sum} (about 7 in FIGURE 5.8).
- Among all intervals $\{[0.05 \cdot (j-1), 0.05 \cdot j] | j = 1, 2, \dots, 80\}$ as conditions of filtration, the selected interval gives the sample with largest size.

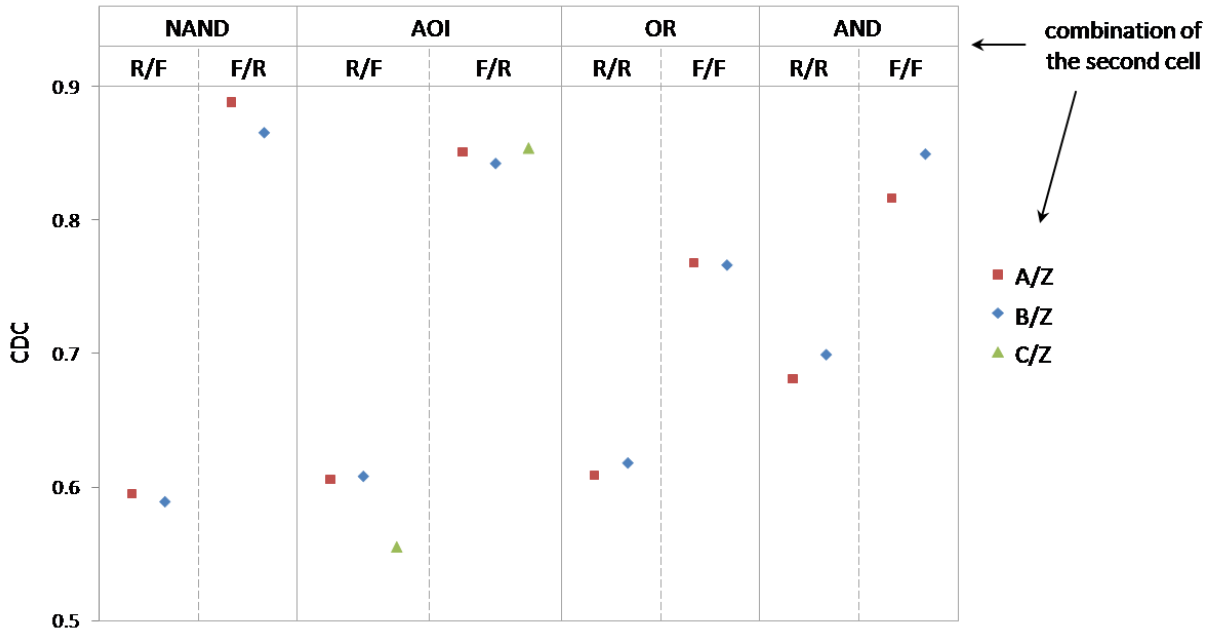
Considering the filtered sample, we focus on CDC coefficients whose combination of the first cell is "$INV - A/Z - F/R$"; as for the combination of the second cell, it may be all possibilities limited to the four cell types: $NAND$, AOI[1], OR, AND. In addition, for each couple of combinations, e.g. "$INV - A/Z - F/R$" (first cell) and "$NAND - A/Z - R/F$" (second cell), we compute the average of all corresponding CDC coefficients. The result is shown in FIGURE 5.9(a). Following the same procedure, we change the combination of the first cell to "$OR - B/Z - F/F$" and obtain FIGURE 5.9(b). From these two figures, we conclude that:

a) CDCs change along with cell type. In FIGURE 5.9(a), for the same I/O pin "A/Z" and I/O edge "R/R", the difference between CDCs of cell OR and cell AND is about 0.1.

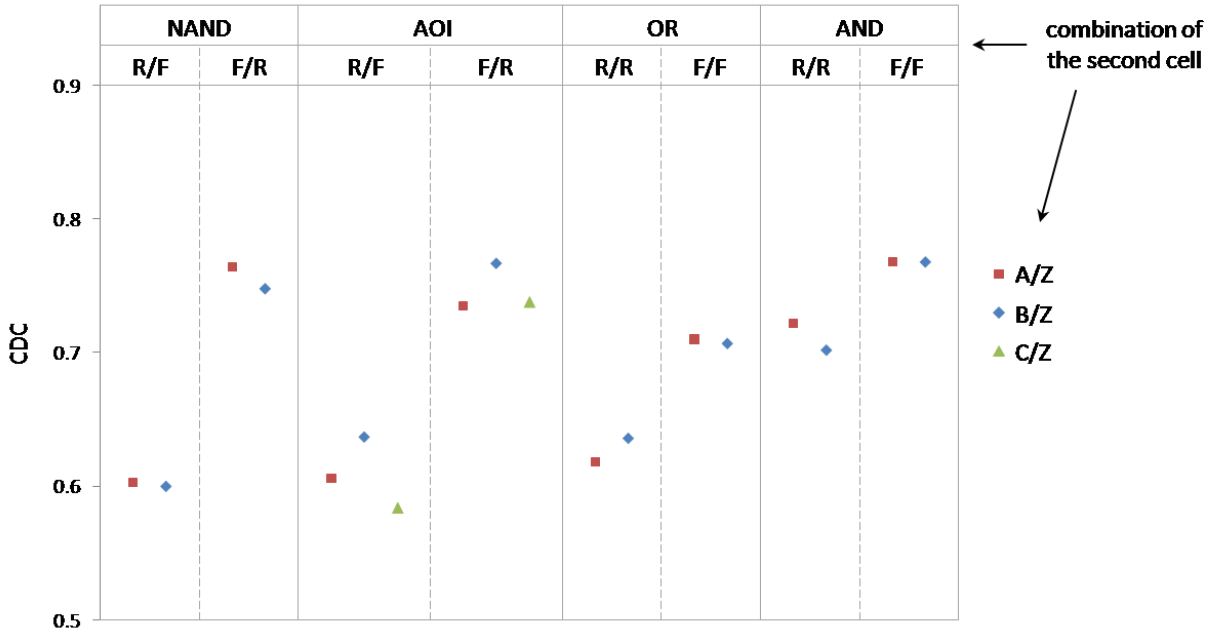
b) Effect of I/O pin is not obvious. In both FIGURES 5.9(a) and (b), CDCs of same cell, I/O edge and different I/O pins are close.

c) Effect of I/O edge is significant. In FIGURE 5.9(a), all combinations of the second cell with the I/O edge "F/R", which is identical to the I/O edge of the first one, have high CDCs. On the contrary, those CDCs of couples with different I/O edges, including "R/F", "R/R" and "F/F", are relatively low. These three cases show a decreasing trend of CDCs following the order "F/F", "R/R" and "R/F". Comparing this order with the I/O edge of the first cell "F/R" shows that effect of input edge is more important than that of output edge. This is also supported in FIGURE 5.9(b) by the fact that CDC with combination of the second cell "$NAND - A/Z - F/R$" is higher than that of "$NAND - A/Z - R/F$" knowing that the I/O edge of the first cell is "F/F".

[1] A compound cell with three input pins



(a) "$INV - A/Z - F/R$" (combination of the first cell)



(b) "$OR - B/Z - F/F$" (combination of the first cell)

FIGURE 5.9 Effects of fixed factors on CDCs

5.4 SUMMARY

This chapter discusses essential applications of our SSTA engine and comparisons of SSTA and CTA. TABLE 5.1 shows the delay gain of SSTA with respect to CTA: about 13% and 25% respectively in the 130 nm and 65 nm technology. The gain is predicted to be more and more significant as the feature size continues to shrink, which implies that SSTA is a promising timing tool. SECTION 5.2 interprets that the discrepancy between orderings obtained respectively by SSTA and CTA comes from two factors: cell-to-cell delay correlation and standard deviation of cell delay. In SECTION 5.3, a study is performed and feeds to conclude that CDCs increase linearly with the compound ratio and effect of I/O edge is significant.

CHAPTER

6

CONCLUSIONS AND FUTURE WORK

In SECTION 6.1, we review the objective of this research and show how the proposed SSTA framework achieved this objective. The main results of our research are also summarized. SECTION 6.2 closes this thesis by making suggestions for future work.

6.1 CONCLUSIONS

Corner-based Timing Analysis (CTA) becomes more and more pessimistic along with the ever shrinking feature size. This trend has resulted in the rapid development of ***Statistical Static Timing Analysis*** (SSTA) in recent years. However, this new generation of timing analysis, both parametric and Monte Carlo methods, has not yet been widely adopted in the industry. On one hand, MC-based methods are accurate, but suffer from the very high computational cost. On the other hand, parametric methods require very little runtime whereas industry and researchers are doubtful of their accuracy due to various weaknesses. The objective of the research was to propose a SSTA framework which performs as fast as parametric methods while not losing too much accuracy compared to MC simulations.

The path-based SSTA framework proposed in this thesis computes path delay distributions by propagating iteratively mean and variance of cell delay with the help of conditional moments. These moments, conditioned on input slope and output load, are stored in a statistical timing library. Compared to existing parametric methods, this semi-MC framework may:

a) avoid cell delay modeling errors;
b) take into account the effects on cell delay: input pin, output edge, input slope, and output load;
c) deal with a large number of process parameters having any type of distribution.

The main difficulty of the SSTA framework is the construction of the statistical timing library. The accuracy of conditional moments in the library is improved by using input signal based on log-logistic distributions and inverters as output load to do timing characterization. In addition, the runtime of characterization could be greatly decreased by the reducing dimension technique, which will be validated in the near future.

From the point of view of accuracy, the SSTA engine allows us to estimate path delay means and standard deviations with relative errors respectively less than 5% and 10%. As for CPU time, it is about 10^5 times faster than a 1500 runs MC simulation for the same path. These figures show that our research objective has been reached.

Also, compared to results of CTA, our SSTA engine has about 13% and 25% of delay gains respectively in 130 nm and 65 nm technology. Such gains will be more significant in the following generations of technology. Another comparison with CTA is about orderings of critical paths. The discrepancy of orderings obtained respectively by SSTA and CTA comes from two factors: cell-to-cell delay correlation and standard deviation of cell delay. The study of cell-to-cell correlation leads to the conclusion that this statistical term increases linearly with the compound ratio r_{sum} and is affected by I/O edge.

6.2 FUTURE WORK

The SSTA framework proposed in this thesis provides acceptable results and runs much faster than MC simulations. However, some work could be done to improve its accuracy and reduce CPU time. This includes:

a) As mentioned in CHAPTER 1, environmental variations are time-varying. Thus, the accuracy of the SSTA engine could be improved by taking into account effects of supply voltage and temperature variations.
b) In addition to mean and variance, it is possible to propagate skewness of cell delay distributions. Then, path delays could be assumed to follow, for example, the skew-Normal distributions. This allows the use of the skew-Normal based MAX approximation in [32], which would provide better accuracy on computation of circuit delay.
c) As for CPU time, the promising acceleration technique in SECTION 4.2 should be validated and applied.

As stated in CHAPTER 2, SSTA must move beyond pure timing analysis to yield analysis and optimization of circuit design to be truly useful for the designers. For this purpose, the problem of optimizing circuit designs with delay correlations, apart from the initial work presented in CHAPTER 5, should be further addressed.

Finally, the proposed SSTA framework has been validated using MC simulations as reference. To be adopted by industry, this framework should be tested with real circuits.

Appendix

A

List of Equations

A.1 EQUATIONS IN CHAPTER 1

$$gd = f_{type,pin,edge}(P, T, V_{dd}, \tau_{in}, C_{out}) \tag{1.1}$$

$$cd_{A_i,Z_j,\gamma_{in}} = h(gd_1, gd_2, \ldots, gd_K) \tag{1.2}$$

$$gd_{[CLK_0 \to CLK_{A_1}]} + gd_{[CLK_{A_1} \to A_1]} + cd_{A_1,Z_1,\gamma_{in}} < gd_{[CLK_0 \to CLK_{Z_1}]} - d_{setup} + T_{CLK} \tag{1.3}$$

$$gd_{[CLK_0 \to CLK_{A_1}]} + gd_{[CLK_{A_1} \to A_1]} + cd_{A_1,Z_1,\gamma_{in}} > gd_{[CLK_0 \to CLK_{Z_1}]} + d_{hold} \tag{1.4}$$

$$SS_{A_i,Z_j,\gamma_{in}} \stackrel{\text{def}}{=} \left\{gd_{\left[CLK_0 \to CLK_{A_i}\right]} + gd_{\left[CLK_{A_i} \to A_i\right]} + cd_{A_i,Z_j,\gamma_{in}}\right\} - \left\{gd_{\left[CLK_0 \to CLK_{Z_j}\right]} - d_{setup}\right\} \tag{1.5}$$

$$HS_{A_i,Z_j,\gamma_{in}} \stackrel{\text{def}}{=} \left\{gd_{\left[CLK_0 \to CLK_{A_i}\right]} + gd_{\left[CLK_{A_i} \to A_i\right]} + cd_{A_i,Z_j,\gamma_{in}}\right\} - \left\{gd_{\left[CLK_0 \to CLK_{Z_j}\right]} + d_{hold}\right\} \tag{1.6}$$

$$SS_{A_i,Z_j,\gamma_{in}} < T_{CLK} \tag{1.7}$$

$$HS_{A_i,Z_j,\gamma_{in}} > 0 \tag{1.8}$$

$$Pr\left\{\bigcap_{i=1}^{I}\bigcap_{j=1}^{J}\bigcap_{\gamma_{in} \in \Gamma_{A_i,Z_j}}\left[\left(SS_{A_i,Z_j,\gamma_{in}} < T_{CLK}\right) \cap \left(HS_{A_i,Z_j,\gamma_{in}} > 0\right)\right]\right\} \geq \theta \tag{1.9}$$

$$\begin{cases} 1 - F_l(p_{upr,l}) = Pr(p_l \geq p_{upr,l}) = \dfrac{\beta}{2} \\ F_l(p_{lwr,l}) = Pr(p_l \leq p_{lwr,l}) = \dfrac{\beta}{2} \end{cases} \tag{1.10}$$

$$Pr\left\{\left(\max_{\gamma_{in} \in \mathbb{\Gamma}^*}\left(SS_{A_i,Z_j,\gamma_{in}}\right) < T_{CLK}\right) \cap \left(\min_{\gamma_{in} \in \mathbb{\Gamma}^*}\left(HS_{A_i,Z_j,\gamma_{in}}\right) > 0\right)\right\} \geq \theta \tag{1.11}$$

$$t_{G_1} = max\left(t_{A_1} + gd_{A_1,G_1}, t_{A_2} + gd_{A_2,G_1}\right) \tag{1.12}$$

$$gd = \sum_{l=1}^{L} p_l \tag{1.13}$$

$$\begin{cases} \mu_{gd} = \sum_{l=1}^{L} \mu_{p_l} = L \cdot \mu_{p_1} \\ \sigma_{gd} = \sqrt{\sum_{l=1}^{L} \sigma_{p_l}^2} = \sqrt{L} \cdot \sigma_{p_1} \end{cases} \tag{1.14}$$

$$w_{gd} = \sum_{l=1}^{L} \left(\mu_{p_l} + 3 \cdot \sigma_{p_l}\right) = L \cdot \mu_{p_1} + 3L \cdot \sigma_{p_1} \tag{1.15}$$

$$\omega = \frac{w_{gd} - \left(\mu_{gd} + 3 \cdot \sigma_{gd}\right)}{\mu_{gd}} = \frac{3\left(L - \sqrt{L}\right) \cdot \sigma_{p_1}}{L \cdot \mu_{p_1}} = 3(1 - L^{-0.5}) \cdot \frac{\sigma_{p_1}}{\mu_{p_1}} \tag{1.16}$$

$$\lim_{L \to +\infty} Pr\left(gd > w_{gd}\right) = \lim_{L \to +\infty} Pr\left(gd > \left(\mu_{gd} + 3\sqrt{L} \cdot \sigma_{gd}\right)\right) = 0 \tag{1.17}$$

$$w_{gd} = \mu_{p_1} + 3 \cdot \sigma_{p_1} = \mu_{gd} + 3 \cdot \sigma_{gd} \tag{1.18}$$

$$w_{pd} = \sum_{k=1}^{K} w_{gd_k} = \sum_{k=1}^{K} \left(\mu_{gd_k} + 3 \cdot \sigma_{gd_k}\right) = \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \sqrt{\sum_{k=1}^{K} \sum_{m=1}^{K} 1 \cdot \sigma_{gd_k} \sigma_{gd_m}} \tag{1.19}$$

$$\mu_{pd} + 3 \cdot \sigma_{pd} = \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \sqrt{\sum_{k=1}^{K} \sum_{m=1}^{K} \rho_{km} \cdot \sigma_{gd_k} \sigma_{gd_m}} \tag{1.20}$$

$$s_i = \mu_i + 3\sigma_i \qquad (i = 1, 2) \tag{1.21}$$

$$w_2 - s_2 > w_1 - s_1 \tag{1.22}$$

A.2 Equations in Chapter 2

$$T_{ILD} = T_{ILD,nom} + \Delta T_{ILD,inter} + \Delta T_{ILD,intra} \tag{2.1}$$

$$\begin{cases} \Delta T_{ILD,inter\,,k_1} = \Delta T_{ILD,inter\,,k_2} \\ cor\left(\Delta T_{ILD,intra\,,k_1}, \Delta T_{ILD,intra\,,k_2}\right) = 0 \end{cases} \tag{2.2}$$

$$T_{ILD} = T_{ILD,nom} + \Delta T_{ILD,inter} + \Delta T_{ILD,spl} + \Delta T_{ILD,ran} \tag{2.3}$$

$$\begin{cases} \Delta T_{ILD,spl,k_1} = \Delta T_{ILD,spl,k_2} \\ cor\left(\Delta T_{ILD,spl,k_1}, \Delta T_{ILD,spl,k_3}\right) \approx 1 \\ cor\left(\Delta T_{ILD,spl,k_1}, \Delta T_{ILD,spl,k_4}\right) \approx 0 \end{cases} \tag{2.4}$$

$$\begin{cases} \Delta T_{ILD,spl,k_1} = \Delta T_{ILD,0,1} + \Delta T_{ILD,1,1} + \Delta T_{ILD,2,1} \\ \Delta T_{ILD,spl,k_2} = \Delta T_{ILD,0,1} + \Delta T_{ILD,1,4} + \Delta T_{ILD,2,11} \end{cases} \tag{2.5}$$

$$gd \approx gd_{nom} + \sum_{l=1}^{L} a_l \cdot \Delta p_l \tag{2.6}$$

$$gd \approx gd_{nom} + \sum_{l=1}^{L} a_l \cdot \Delta p_l + \sum_{l=1}^{L} b_l \cdot \Delta p_l^2 + \sum_{\forall l_1 \neq l_2}^{L} c_{l_1 l_2} \cdot \Delta p_{l_1} \Delta p_{l_2} \tag{2.7}$$

$$\begin{cases} \mu_Z = \mu_X + \mu_Y \\ \sigma_Z^2 = \sigma_X^2 + \sigma_Y^2 + \rho_{XY} \cdot \sigma_X \sigma_Y \end{cases} \tag{2.8}$$

$$\begin{cases} \varphi(x) = \dfrac{1}{\sqrt{2\pi}} \cdot e^{\frac{-x^2}{2}} \\ \Phi(x) = \displaystyle\int_{-\infty}^{x} \varphi(u) du \end{cases} \tag{2.9}$$

$$\widehat{W} = \Phi\left(\frac{\mu_V}{\sigma_V}\right) \cdot X + \left(1 - \Phi\left(\frac{\mu_V}{\sigma_V}\right)\right) \cdot Y + \varphi\left(\frac{\mu_V}{\sigma_V}\right) \cdot \sigma_V \tag{2.10}$$

$$\begin{cases} \mu_V = \mu_X - \mu_Y \\ \sigma_V = (\sigma_X^2 + \sigma_Y^2 - \rho_{XY} \cdot \sigma_X \sigma_Y)^{1/2} \end{cases} \tag{2.11}$$

A.3 EQUATIONS IN CHAPTER 3

$$p_l = p_{nom,l} + \Delta p_{inter,l} + \Delta p_{intra,l} \tag{3.1}$$

$$cd = \max(pd_1, pd_2, \dots, pd_N) \tag{3.2}$$

$$E(X|Y = y) = \int_{-\infty}^{\infty} x \cdot f(x|y) dx \tag{3.3}$$

$$Var(X|Y = y) = E(X^2|Y = y) - E^2(X|Y = y) \tag{3.4}$$

$$\begin{cases} \mu_X = E(X) = E[E(X|Y = y)] \\ \sigma_X^2 = Var(X) = E[Var(X|Y = y)] + Var[E(X|Y = y)] \end{cases} \tag{3.5}$$

$$\begin{cases} Pr(Y = y_i) = \alpha_i > 0 \qquad i = 1, \dots, I \\ \sum_{i=1}^{I} \alpha_i = 1 \end{cases} \tag{3.6}$$

$$\begin{cases} \mu_X = \sum_{i=1}^{I} \alpha_i \cdot E(X|Y = y_i) \\ \sigma_X^2 = \sum_{i=1}^{I} \alpha_i \cdot \{Var(X|Y = y_i) + [E(X|Y = y_i) - E(X)]^2\} \end{cases} \tag{3.7}$$

$$\begin{cases} \mu_X = \int E(X|Y = y) \cdot f(y)dy \\ \sigma_X^2 = \int \{Var(X|Y = y) + [E(X|Y = y) - \mu_X]^2\} \cdot f(y)dy \end{cases} \tag{3.8}$$

$$\begin{cases} E(\tau_{out}|\tau_6, c) \approx \frac{c_3 - c}{c_3 - c_2} \cdot E(\tau_{out}|\tau_6, c_2) + \frac{c - c_2}{c_3 - c_2} \cdot E(\tau_{out}|\tau_6, c_3) \\ E(\tau_{out}|\tau_7, c) \approx \frac{c_3 - c}{c_3 - c_2} \cdot E(\tau_{out}|\tau_7, c_2) + \frac{c - c_2}{c_3 - c_2} \cdot E(\tau_{out}|\tau_7, c_3) \end{cases} \tag{3.9}$$

$$E(\tau_{out}|\tau, c) \approx \frac{\tau_7 - \tau}{\tau_7 - \tau_6} \cdot E(\tau_{out}|\tau_6, c) + \frac{\tau - \tau_6}{\tau_7 - \tau_6} \cdot E(\tau_{out}|\tau_7, c) \tag{3.10}$$

$$y_i = \frac{s_{i-1} + s_i}{2} \qquad i = 1, \dots, I \tag{3.11}$$

$$\alpha_i = \begin{cases} \int_{-\infty}^{s_1} f(\tau_{in})d\tau_{in} & i = 1 \\ \int_{s_{i-1}}^{s_i} f(\tau_{in})d\tau_{in} & i = 2, \dots, I-1 \\ \int_{s_{I-1}}^{+\infty} f(\tau_{in})d\tau_{in} & i = I \end{cases} \tag{3.12}$$

$$\begin{cases} \mu_{\tau_{out}} = \sum_{i=1}^{I} \alpha_i \cdot E(\tau_{out}|y_i, c) \\ \sigma_{\tau_{out}}^2 = \sum_{i=1}^{I} \alpha_i \cdot \left\{Var(\tau_{out}|y_i, c) + \left[E(\tau_{out}|y_i, c) - \mu_{\tau_{out}}\right]^2\right\} \end{cases} \tag{3.13}$$

$$\begin{cases} \mu_{gd} = \sum_{i=1}^{I} \alpha_i \cdot E(gd|y_i, c) \\ \sigma_{gd}^2 = \sum_{i=1}^{I} \alpha_i \cdot \left\{ Var(gd|y_i, c) + \left[E(gd|y_i, c) - \mu_{gd} \right]^2 \right\} \end{cases} \tag{3.14}$$

$$\begin{cases} E(\tau_{out}|\tau_{in}, c) = b_1 + b_2 \cdot \tau_{in} \\ Var(\tau_{out}|\tau_{in}, c) = b_3 + b_4 \cdot \tau_{in} \end{cases} \tag{3.15}$$

$$\mu_{\tau_{out}} = b_1 + b_2 \cdot \mu_{\tau_{in}} \tag{3.16}$$

$$\sigma_{\tau_{out}}^2 = \left(b_3 + b_4 \cdot \mu_{\tau_{in}} \right) + \left(b_2 \cdot \sigma_{\tau_{in}} \right)^2 \tag{3.17}$$

$$E(\tau_{out}|\tau_{in}, c) = \frac{\tau_7 \cdot E(\tau_{out}|\tau_6, c) - \tau_6 \cdot E(\tau_{out}|\tau_7, c)}{\tau_7 - \tau_6} + \frac{E(\tau_{out}|\tau_7, c) - E(\tau_{out}|\tau_6, c)}{\tau_7 - \tau_6} \cdot \tau_{in} \tag{3.18}$$

$$Var(\tau_{out}|\tau_{in}, c) = \frac{\tau_7 \cdot Var(\tau_{out}|\tau_6, c) - \tau_6 \cdot Var(\tau_{out}|\tau_7, c)}{\tau_7 - \tau_6} + \frac{Var(\tau_{out}|\tau_7, c) - Var(\tau_{out}|\tau_6, c)}{\tau_7 - \tau_6} \cdot \tau_{in} \tag{3.19}$$

$$\begin{cases} b_1 = \dfrac{\tau_7 \cdot E(\tau_{out}|\tau_6, c) - \tau_6 \cdot E(\tau_{out}|\tau_7, c)}{\tau_7 - \tau_6} \\ b_2 = \dfrac{E(\tau_{out}|\tau_7, c) - E(\tau_{out}|\tau_6, c)}{\tau_7 - \tau_6} \\ b_3 = \dfrac{\tau_7 \cdot Var(\tau_{out}|\tau_6, c) - \tau_6 \cdot Var(\tau_{out}|\tau_7, c)}{\tau_7 - \tau_6} \\ b_4 = \dfrac{Var(\tau_{out}|\tau_7, c) - Var(\tau_{out}|\tau_6, c)}{\tau_7 - \tau_6} \end{cases} \tag{3.20}$$

$$\begin{cases} \mu_{pd} = \sum_{k=1}^{K} \mu_{gd_k} \\ \sigma_{pd}^2 = \sum_{k=1}^{K} \sum_{m=1}^{K} \rho_{km} \cdot \sigma_{gd_k} \sigma_{gd_m} \end{cases} \tag{3.21}$$

$$gd_k \approx gd_{nom,k} + a_{1k} \cdot \Delta p_{1k} + a_{2k} \cdot \Delta p_{2k} \tag{3.22}$$

$$cor(gd_1, gd_2) = \frac{cov(gd_1, gd_2)}{\sigma_{gd_1} \sigma_{gd_2}} \tag{3.23}$$

$$cov(gd_1, gd_2) = a_{11} a_{12} \cdot cov(\Delta p_{11}, \Delta p_{12}) + a_{21} a_{22} \cdot cov(\Delta p_{21}, \Delta p_{22}) \tag{3.24}$$

$$\Delta p_{lk} = \Delta p_{inter,lk} + \Delta p_{intra,lk} \qquad (l = 1,2) \tag{3.25}$$

$$cov(\Delta p_{l1}, \Delta p_{l2}) = \sigma_{\Delta p_{inter,l1}} \sigma_{\Delta p_{inter,l2}} \cdot cor\left(\Delta p_{inter,l1}, \Delta p_{inter,l2}\right) + \sigma_{\Delta p_{intra,l1}} \sigma_{\Delta p_{intra,l2}} \cdot cor\left(\Delta p_{intra,l1}, \Delta p_{intra,l2}\right) \qquad (l = 1,2) \tag{3.26}$$

$$\begin{cases} P^{NM} = \left(p_1^{NM}, p_2^{NM}, \dots, p_{n_1}^{NM}\right) \\ P^{PM} = \left(p_1^{PM}, p_2^{PM}, \dots, p_{n_2}^{PM}\right) \\ P^{S} = \left(p_1^{S}, p_2^{S}, \dots, p_{n_3}^{S}\right) \end{cases} \qquad L = n_1 + n_2 + n_3 \tag{3.27}$$

$$gd \approx gd^{NM} + gd^{PM} + gd^{S} \tag{3.28}$$

$$\begin{cases} gd^{NM} = f_{type,pin,edge}\left(P^{NM}, T, V_{dd}, \tau_{in}, C_{out}\right) \\ gd^{PM} = f_{type,pin,edge}\left(P^{PM}, T, V_{dd}, \tau_{in}, C_{out}\right) \\ gd^{S} = f_{type,pin,edge}\left(P^{S}, T, V_{dd}, \tau_{in}, C_{out}\right) \end{cases} \tag{3.29}$$

$$\begin{cases} cor(gd^{NM}, gd^{PM}) = 0 \\ cor(gd^{NM}, gd^{S}) = 0 \\ cor(gd^{PM}, gd^{S}) = 0 \end{cases} \tag{3.30}$$

$$\rho_{km} = \frac{cov(gd_k, gd_m)}{\sigma_{gd_k}\sigma_{gd_m}} \tag{3.31}$$

$$cov(gd_k, gd_m) = cov(gd_k^{NM}, gd_m^{NM}) + cov(gd_k^{PM}, gd_m^{PM}) + cov(gd_k^{S}, gd_m^{S}) \tag{3.32}$$

$$gd^{NM} = gd_{inter}^{NM} + gd_{intra}^{NM} \tag{3.33}$$

$$cor\left(gd_{k,inter}^{NM}, gd_{m,inter}^{NM}\right) \approx 1 \tag{3.34}$$

$$cov(gd_k^{NM}, gd_m^{NM}) \approx \sigma_{gd_k,inter}^{NM} \cdot \sigma_{gd_m,inter}^{NM} \tag{3.35}$$

$$cov(gd_k, gd_m) \approx \sigma_{gd_k,inter}^{NM} \cdot \sigma_{gd_m,inter}^{NM} + \sigma_{gd_k,inter}^{PM} \cdot \sigma_{gd_m,inter}^{PM} + \sigma_{gd_k,inter}^{S} \cdot \sigma_{gd_m,inter}^{S} \tag{3.36}$$

$$cor(pd_1, pd_2) = \frac{cov(pd_1, pd_2)}{\sigma_{pd_1}\sigma_{pd_2}} \tag{3.37}$$

$$cov(pd_1, pd_2) = cov\left(\sum_{k_1=1}^{K_1} gd_{k_1}, \sum_{k_2=1}^{K_2} gd_{k_2}\right) = \sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2} cov\left(gd_{k_1}, gd_{k_2}\right) \tag{3.38}$$

$$cov\left(gd_{k_1}, gd_{k_2}\right) = \sigma_{gd_{k_1}} \cdot \sigma_{gd_{k_2}} \tag{3.39}$$

$$cov\left(gd_{k_1}, gd_{k_2}\right) \approx \sigma_{gd_{k_1},inter}^{NM} \cdot \sigma_{gd_{k_2},inter}^{NM} + \sigma_{gd_{k_1},inter}^{PM} \cdot \sigma_{gd_{k_2},inter}^{PM} + \sigma_{gd_{k_1},inter}^{S} \cdot \sigma_{gd_{k_2},inter}^{S} \tag{3.40}$$

$$\begin{cases} e_{abs} = |\hat{\rho} - \rho| \\ e_{rel} = \dfrac{|\hat{\rho} - \rho|}{\rho}\% \end{cases} \tag{3.41}$$

$$\begin{cases} p_{oe} = \dfrac{1}{N} \cdot \sum_{i=1}^{N} if\{\hat{\rho}_i - \rho_i < 0, then\ 1, else\ 0\} \\ p_{ue} = \dfrac{1}{N} \cdot \sum_{i=1}^{N} if\{\hat{\rho}_i - \rho_i > 0, then\ 1, else\ 0\} \\ p_{abs} = \dfrac{1}{N} \cdot \sum_{i=1}^{N} if\{e_{abs} \leq 0.2, then\ 1, else\ 0\} \\ p_{rel} = \dfrac{1}{N} \cdot \sum_{i=1}^{N} if\{e_{rel} \leq 20\%, then\ 1, else\ 0\} \end{cases} \tag{3.42}$$

$$pd_{data} - pd_{clk} < T_{CLK} \tag{3.43}$$

$$Var(pd_{data} - pd_{clk}) = \sigma_{data}^2 + \sigma_{clk}^2 - 2 \cdot \rho_{dc} \cdot \sigma_{data} \cdot \sigma_{clk} \tag{3.44}$$

A.4 EQUATIONS IN CHAPTER 4

$$F(x; \alpha, \beta) = \left[\left(\frac{\alpha}{x}\right)^{\beta} + 1\right]^{-1} \qquad (x > 0) \tag{4.1}$$

$$\begin{cases} \tau_{in} = \dfrac{5}{3} \cdot (t_2 - t_1) \\ \Delta V = |0 - V_{min}| = |V_{min}| \\ \Delta t = t_{min} - t_0 \end{cases} \tag{4.2}$$

$$\widehat{V} = \widehat{H}(t) = \begin{cases} -\frac{\Delta V}{\Delta t} \cdot (t + \Delta t - t_{min}) & (t \leq t_{min}) \\ (V_{dd} + \Delta V) \cdot \left\{ \left[\frac{\alpha}{(t - t_{min})/\tau_{in}} \right]^{\beta} + 1 \right\}^{-1} - \Delta V & (t > t_{min}) \end{cases} \tag{4.3}$$

$$\begin{cases} \Delta V = g_{\Delta V}(\tau_{in}) \\ \Delta t = g_{\Delta t}(\tau_{in}) \\ \beta = g_{\beta}(\tau_{in}) \end{cases} \tag{4.4}$$

$$\begin{cases} \Delta V = \dfrac{C_{\Delta V}}{A_{\Delta V} + B_{\Delta V} \cdot \tau_{in}} \\ \Delta t = A_{\Delta t} + B_{\Delta t} \cdot \tau_{in} \\ \beta = \dfrac{C_{\beta}}{A_{\beta} + B_{\beta} \cdot \tau_{in}} + D_{\beta} \end{cases} \tag{4.5}$$

$$t = t_{min} + \tau_{in} \cdot \alpha \cdot \left(\frac{V_{dd} - \widehat{V}}{\widehat{V} + \Delta V} \right)^{-\frac{1}{\beta}} \qquad (t > t_{min}) \tag{4.6}$$

$$t_2 - t_1 = \tau_{in} \cdot \alpha \cdot \left[\left(\frac{0.2}{0.8 + \frac{\Delta V}{V_{dd}}} \right)^{-\frac{1}{\beta}} - \left(\frac{0.8}{0.2 + \frac{\Delta V}{V_{dd}}} \right)^{-\frac{1}{\beta}} \right] = 0.6 \cdot \tau_{in} \tag{4.7}$$

$$\alpha = 0.6 \cdot \left[\left(\frac{0.2}{0.8 + \frac{\Delta V}{V_{dd}}} \right)^{-\frac{1}{\beta}} - \left(\frac{0.8}{0.2 + \frac{\Delta V}{V_{dd}}} \right)^{-\frac{1}{\beta}} \right]^{-1} \tag{4.8}$$

$$\begin{cases} \dfrac{E(\tau_{out}|\tau_{in}, C_{out})}{\mu_{ft}} = h_1\left(\dfrac{\tau_{in}}{\mu_{ft}}\right) \\ \dfrac{Var(\tau_{out}|\tau_{in}, C_{out})}{\sigma_{ft}^2} = h_2\left(\dfrac{\tau_{in}}{\mu_{ft}}\right) \\ \dfrac{E(gd|\tau_{in}, C_{out})}{\mu_{ft}} = h_3\left(\dfrac{\tau_{in}}{\mu_{ft}}\right) \\ \dfrac{Var(gd|\tau_{in}, C_{out})}{\sigma_{ft}^2} = h_4\left(\dfrac{\tau_{in}}{\mu_{ft}}\right) \end{cases} \tag{4.9}$$

$$\begin{cases} \mu_{ft} = A + B \cdot C_{out} \\ \sigma_{ft}^2 = B^2 \cdot C_{out}^2 \end{cases} \tag{4.10}$$

A.5 Equations in Chapter 5

$$\begin{cases} w_{IPN_m} = \underset{u \in U_{100,m}}{max}\left(w_{pd_u}\right) \\ s_{IPN_m} = \underset{u \in U_{100,m}}{max}\left(\mu_{pd_u} + 3 \cdot \sigma_{pd_u}\right) \end{cases} \tag{5.1}$$

$$r_{GS} = \frac{GS}{S_{CTA}}\% \tag{5.2}$$

$$\overline{GD} = \frac{1}{7} \cdot \sum_{m=1}^{7}\left(w_{IPN_m} - s_{IPN_m}\right) \tag{5.3}$$

$$\overline{r_{GD}} = \frac{1}{7} \cdot \sum_{m=1}^{7} \frac{w_{IPN_m} - s_{IPN_m}}{w_{IPN_m}}\% \tag{5.4}$$

$$s_{pd_{u_i}} = \mu_{pd_{u_i}} + 3 \cdot \sigma_{pd_{u_i}} \tag{5.5}$$

$$rk_{u_i} = \sum_{j=1}^{100} if\left\{w_{pd_{u_j}} \geq w_{pd_{u_i}}, then\ 1, else\ 0\right\} \tag{5.6}$$

$$\begin{cases} rk_{u_{i_1}} = M - k + 1 \\ rk_{u_{i_2}} = M - k + 2 \\ \quad \cdots\cdots \\ rk_{u_{i_k}} = M \end{cases} \tag{5.7}$$

$$w_{gd_k} = \mu_{gd_k} + \theta_k \cdot \sigma_{gd_k} \qquad (k = 1, 2, \ldots, K) \tag{5.8}$$

$$s_{pd} = \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} \rho_{km} \cdot \sigma_{gd_k}\sigma_{gd_m}} \tag{5.9}$$

$$w_{pd} = \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} 1 \cdot \left(\frac{\theta_k \cdot \sigma_{gd_k}}{3}\right)\left(\frac{\theta_m \cdot \sigma_{gd_m}}{3}\right)} \tag{5.10a}$$

$$= \sum_{k=1}^{K} \mu_{gd_k} + 3 \cdot \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} 1 \cdot \left(\frac{w_{gd_k} - \mu_{gd_k}}{3}\right)\left(\frac{w_{gd_k} - \mu_{gd_k}}{3}\right)} \tag{5.10b}$$

$$s_{pd}^{'} = \mu_{pd} + 3 \cdot \sigma_{pd}^{'} \tag{5.11}$$

$$s_{pd}^{''} = \mu_{pd} + 3 \cdot \sigma_{pd}^{''} \tag{5.12}$$

$$\sigma_{pd}^{'} = \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} 1 \cdot \sigma_{gd_k}\sigma_{gd_m}} \tag{5.13}$$

$$\sigma_{pd}^{''} = \sqrt{\sum_{k=1}^{K}\sum_{m=1}^{K} \rho_{km} \cdot \left(\frac{\theta_k \cdot \sigma_{gd_k}}{3}\right)\left(\frac{\theta_m \cdot \sigma_{gd_m}}{3}\right)} \tag{5.14}$$

$$Var(pd_{data} - pd_{clk}) = \sigma_{data}^2 + \sigma_{clk}^2 - 2 \cdot \rho_{dc} \cdot \sigma_{data} \cdot \sigma_{clk} \tag{5.15}$$

$$\begin{cases} r_{sum} = r_1 + r_2 \\ r_{dif} = |r_1 - r_2| \\ r_{pdt} = r_1 \cdot r_2 \end{cases} \tag{5.16}$$

$$\begin{cases} r_1 = \dfrac{\mu_{\tau_{in},1}}{\mu_{\tau_{out},1}} \in [0.85, 0.9] \\ r_2 = \dfrac{\mu_{\tau_{in},2}}{\mu_{\tau_{out},2}} \in [0.85, 0.9] \end{cases} \tag{5.17}$$

Appendix

B

Author's Publications

[P1] V. MIGAIROU, R. WILSON, S. ENGELS, Z. WU, N. AZEMARD, and P. MAURINE, "A Simple Statistical Timing Analysis Flow and its Application to Timing Margins Evaluation", *Proc. PATMOS*, 2007, pages 138 – 147.

[P2] B.REBAUD, M.BELLEVILLE, C.BERNARD, Z .WU, M.ROBERT, P.MAURINE, and N. AZEMARD, "Setup and Hold Timing Violations Induced by Process Variations, in a Digital Multiplier", *Proc. ISVLSI*, 2008, pages 316 – 321.

[P3] Z. WU, P. MAURINE, N. AZEMARD, and G. DUCHARME, "SSTA Considering Effects of Structure Correlations, Input Slope and Output Load Variations", *Proc. FTFC*, 2008, pages 39 – 44.

[P4] B.REBAUD, M.BELLEVILLE, C.BERNARD, Z .WU, M.ROBERT, P.MAURINE, and N. AZEMARD, "Impact de la Variabilité des Caractéristiques Temporelles des Cellules Combinatoires et Séquentielles sur un Opérateur Numérique", *Proc. FTFC*, 2008, pages 45 – 50.

[P5] Z. WU, P. MAURINE, N. AZEMARD, and G. DUCHARME, "SSTA with Correlations Considering Input Slope and Output Load Variations", *Proc. VLSI-SOC*, 2008, pages 164 – 167.

[P6] Z. WU, P. MAURINE, N. AZEMARD, and G. DUCHARME, "Conditional Moments Based SSTA Considering Switching Process Induced Correlations", *Proc. DCIS*, 2008.

[P7] Z. WU, P. MAURINE, N. AZEMARD, and G. DUCHARME, "SSTA Considering Switching Process Induced Correlations", *Proc. APCCAS*, 2008, pages 562 – 565.

[P8] Z. WU, P. MAURINE, N. AZEMARD, and G. DUCHARME, "Interpretation of SSTA Results", *Proc. FTTC*, 2009.

[P9] Z. WU, P. MAURINE, N. AZEMARD, and G. DUCHARME, "Interpreting SSTA Results with Correlations", *Proc. PATMOS*, 2009.

REFERENCES

[1] R. WILSON, "Statistical Timing Analysis Moves from Interesting to Necessary", *Electrical Design News*, June 2006.

[2] D. MALINIAK, "Timing Analysis Rounds the Corner to Statistics", *Electronic Design*, December 2005.

[3] V. MIGAIROU, "Conception et Vérification des Circuits CMOS Digitaux Basées sur les Statistiques : Application à l'Evaluation des Marges Temporelles de Conception", *Thèse*, Chapitre 1, Université de Montpellier II, 2007.

[4] http://en.wikipedia.org/wiki/Bilinear_interpolation

[5] J. MODER, C. Phillips, and E. Davis, *Project Management with CPM, PERT and Precedence Diagramming*, Chapter 9, Van Nostrand Reinhold, 1983.

[6] D. BONING and S. NASSIF, "Models of Process Variations in Device and Interconnect", *Design of High Performance Microprocessor Circuits*, Chapter 6, Wiley-IEEE Press, 2000.

[7] http://www-device.eecs.berkeley.edu/~bsim3/bsim_ent.html

[8] T. KIRKPATRICK and N. CLARK, "PERT as an aid to logic design", *IBM Journal of Research and Development*, vol. 10, no. 2, pages 135 – 141, 1966.

[9] H. JYU, S. MALIK, S. DEVADAS, and K. KEUTZER, "Statistical Timing Analysis of Combinational Logic Circuits", *IEEE Trans. VLSI Systems*, vol. 1, no. 2, pages 126 – 137, 1993.

[10] R. BRAWHEAR, N. MENEZES, C. OH, L. PILLAGE, and M. MERCER, "Predicting Circuit Performance Using Circuit-level Statistical Timing Analysis", *Proc. DATE*, 1994, pages 332 – 337.

[11] H. CHANG and S. SAPATNEKAR, "Statistical Timing Analysis Considering Spatial Correlations Using a Single PERT-like Traversal", *Proc. ICCAD*, 2003, pages 621 – 625.

[12] A. AGARWAL, D. BLAAUW, and V. ZOLOTOV, "Statistical Timing Analysis for Intra-die Process Variations with Spatial Correlations", *Proc. ICCAD*, 2003, pages 900 – 907.

[13] C. VISWESWARIAH, K. RAVINDRAN, K. KALAFALA, S. WALER, and S. NARAYAN, "First-order Incremental Block-based Statistical Timing Analysis", *Proc. DAC*, 2004, pages 331 – 336.

[14] L. ZHANG, W. CHEN, Y. HU, J. GUBNER, and C. CHEN, "Correlation-preserved non-Gaussian Statistical Timing Analysis with Quadratic Timing Model", *Proc. DAC*, 2005, pages 83 – 88.

[15] Y. ZHAN, A. STROJWAS, X. LI, and L. PILEGGI, "Correlation-aware Statistical Timing Analysis with non-Gaussian Delay Distribution", *Proc. DAC*, 2005, pages 77 – 82.

[16] V. KHANDELWAL and A. SRIVASTAVA, "A General Framework for Accurate Statistical Timing Analysis Considering Correlations", *Proc. DAC*, 2005, pages 89 – 94.

[17] H. CHANG, V. ZOLOTOV, S. NARAYAN, and C. VISWESWARIAH, "Parameterized Block-based Statistical Timing Analysis with non-Gaussian Parameters, Nonlinear Delay Functions", *Proc. DAC*, 2005, pages 71 – 76.

[18] J. SINGH and S. SAPATNEKAR, "Statistical Timing Analysis with Correlated non-Gaussian Parameters Using Independent Component Analysis", *Proc. DAC*, 2006, pages 155 – 160.

[19] L. CHENG, J. XIONG, and L. HE, "Non-Linear Statistical Static Timing Analysis for non-Gaussian Variation Sources", *Proc. DAC*, 2007, pages 250 – 255.

[20] Z. FENG, P. LI, and Y. ZHAN, "Fast Second-order Statistical Static Timing Analysis Using Parameter Dimension Reduction", *Proc. DAC*, 2007, pages 244 – 249.

[21] F. NAJM and N. MENEZES, "Statistical Timing Analysis Based on a Timing Yield Model", *Proc. DAC*, 2004, pages 460 – 465.

[22] C. AMIN, N. MENEZES, K. KILLPACK, F. DARTU, U. CHOUDHURY, N. HAKIM, and Y. ISMAIL, "Statistical Static Timing Analysis: How simple can we get?", *Proc. DAC*, 2005, pages 652 – 657.

[23] K. HELOUE and F. NAJM, "Statistical Timing Analysis with Two-sided Constraints", *Proc. ICCAD*, 2005, pages 829 – 836.

[24] L. SCHEFFER, "The Count of Monte Carlo", *Proc. TAU*, 2004.

[25] S. TASIRAN and A. DEMIR, "Smart Monte Carlo for Yield Estimation", *Proc. TAU*, 2006.

[26] R. KANJ, R. JOSHI, and S. NASSIF, "Mixture Importance Sampling and its Application to the Analysis of SRAM Designs in the Presence of Rare Failure Events", *Proc. DAC*, 2006, pages 69 – 72.

[27] V. VEETIL, D. BLAAUW, and D. SYLVESTER, "Criticality Aware Latin Hypercube Sampling for Efficient Statistical Timing Analysis", *Proc. TAU*, 2007.

[28] M. BUHLER, J. KOEHL, J. BICKFORD, J. HIBBELER, U. SCHLICHTMANN, R. SOMMER, M. PRONATH, and A. RIPP, "DATE 2006 Special Session: DFM/DFY Design for Manufacturability and Yield - Influence of Process Variations in Digital, Analog and Mixed-signal Circuit Design", *Proc. DATE*, 2006, pages 1 – 6.

[29] D. BLAAUW, K. CHOPRA, A. SRIVASTAVA, and L. SCHEFFER, "Statistical Timing Analysis: From Basic Principles to State of the Art", *IEEE Trans. CAD*, vol. 27, no. 4, pages 589 – 607, 2008.

[30] C. CLARK, "The Greatest of a Finite Set of Random variables", *Journal Operation Research*, vol. 9, no. 2, pages 145 – 162, 1961.

[31] L. XIE, A. DAVOODI, J. ZHANG, and T. WU, "Adjustment-based Modeling for Statistical Static Timing Analysis with High Dimension of Variability", *Proc. ICCAD*, 2008, pages 181 – 184.

[32] K. CHOPRA, B. ZHAI, D. BLAAUW, and D. SYLVESTER, "A New Statistical MAX Operation for Propagating Skewness in Statistical Timing Analysis", *Proc. ICCAD*, 2006, pages 237 – 243.

[33] A. AGARWAL, F. DARTU, and D. BLAAUW, "Statistical Gate Delay Model Considering Multiple Input Switching", *Proc. DAC*, 2004, pages 658 – 663.

[34] Y. KUMAR, J. LI, C. TALARICO, and J. WANG, "A Probabilistic Collocation Method Based Statistical Gate Delay Model Considering Process Variations and Multiple Input Switching", *Proc. DATE*, 2005, pages 770 – 775.

[35] M. AGARWAL, K. AGARWAL, D. SYLVESTER, and D. BLAAUW, "Statistical Modeling of Cross-coupling Effects in VLSI Interconnects", *Proc. ASP-DAC*, 2005, pages 503 – 506.

[36] R. CHEN and H. ZHOU, "Clock Schedule Verification under Process Variations", *Proc. ICCAD*, 2004, pages 619 – 625.

[37] L. ZHANG, Y. HU, and C. CHEN, "Statistical Timing Analysis in Sequential Circuit for On-chip Global Interconnect Pipelining", *Proc. DAC*, 2004, pages 904 – 907.

[38] K. CHOPRA, S. SHAH, A. SRIVASTAVA, D. BLAAUW, and D. SYLVESTER, "Parametric Yield Maximization Using Gate Sizing Based on Efficient Statistical Power and Delay Gradient Computation", *Proc. ICCAD*, 2005, pages 1023 – 1028.

[39] J. XIONG, V. ZOLOTOV, N. VENKATESWARAN, and C. VISWESWARIAH, "Criticality Computation in Parameterized Statistical Timing", *Proc. DAC*, 2006, pages 63 – 68.

[40] M. GUTHAUS, N. VENKATESWARAN, C. VISWESWARIAH, and V. ZOLOTOV, "Gate Sizing Using Incremental Parameterized Statistical Timing Analysis", *Proc. ICCAD*, 2005, pages 1029 – 1036.

[41] D. SINHA, N. SHENOY, and H. ZHOU, "Statistical Gate Sizing for Timing Yield Optimization", *Proc. ICCAD*, 2005, pages 1140 – 1146.

[42] V. KHANDELWAL, A. DAVOODI, A. NANAVATI, and A. SRIVASTAVA, "A Probabilistic Approach to Buffer Insertion", *Proc. ICCAD*, 2003, pages 560 – 567.

[43] http://www.synopsys.com/

[44] S. YEN, D. DU, and S. GHANTA, "Efficient Algorithms for Extracting the K Most Critical Paths in Timing Analysis", *Proc. DAC*, 1989, pages 649 – 654.

[45] http://www.cadence.com/

[46] D. SINHA, H. ZHOU, and N. SHENOY, "Advances in Computation of the Maximum of a Set of Random Variables", *Proc. ISQED*, 2006, pages 306 – 311.

[47] A. SRIVASTAVA, D. SYLVESTER, D. BLAAUW, *Statistical Analysis and Optimization of VLSI: Timing and Power*, Chapter 3, Springer, 2005.

[48] P. BILLINGSLEY, *Probability and Measure*, 2nd edition, page 477, Wiley New York, 1986.

[49] http://www.r-project.org/

[50] B. LASBOUYGUES, “Analyse Statique Temporelle des Performances en Présence de Variations de Tension d’Alimentation et de Température”, *Thèse*, Chapitre 2, Université de Montpellier II, 2006.

[51] http://en.wikipedia.org/wiki/Least_squares_method

[52] P. MAURINE, “Modélisation et Optimisation des Performances de la Logique Statique en Technologie CMOS Submicronique”, *Thèse*, Chapitre 3, Université de Montpellier II, 2001.

SSTA Framework Based on Moments Propagation

Abstract

Corner-based Timing Analysis (CTA) becomes more and more pessimistic along with the shrinking feature size. This trend has urged the need of ***Statistical Static Timing Analysis*** (SSTA). However, this new generation of timing analysis has not yet widely adopted in the industry due to various weaknesses. The path-based SSTA framework proposed in this thesis computes path delay distributions by propagating iteratively mean and variance of cell delay with the help of conditional moments. These moments, conditioned on input slope and output load, are stored in a statistical timing library. This framework performs as fast as parametric methods while not losing too much accuracy compared to Monte Carlo simulations, which meets the objective of the research. Another contribution of this thesis is the improvement of the techniques to do timing characterization. We use input signals based on log-logistic distributions and inverters as output load to capture slope and load variations. In addition, the runtime of characterization could be greatly saved by the reducing dimension technique, which would be validated in the near future. In the part of applications, our SSTA engine shows significant delay gains with respect to CTA. The discrepancy of critical paths orderings obtained respectively by SSTA and CTA is explained as well. Finally, a study of cell-to-cell correlation is given.