

10-1-1995

# Statistical Estimation of Combinational and Sequential CMOS Digital Circuit Activity Considering Uncertainty of Gate Delay Models

Tan Li Chou

*Purdue University School of Electrical and Computer Engineering*

Kaushik Roy

*Purdue University School of Electrical and Computer Engineering*

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

---

Chou, Tan Li and Roy, Kaushik, "Statistical Estimation of Combinational and Sequential CMOS Digital Circuit Activity Considering Uncertainty of Gate Delay Models" (1995). *ECE Technical Reports*. Paper 146.

<http://docs.lib.purdue.edu/ecetr/146>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

---

STATISTICAL ESTIMATION OF  
COMBINATIONAL AND SEQUENTIAL  
CMOS DIGITAL CIRCUIT ACTIVITY  
CONSIDERING UNCERTAINTY OF  
GATE DELAY MODELS

TAN-LI CHOU  
KAUSHIK ROY

TR-ECE 95-25  
OCTOBER 1995



SCHOOL OF ELECTRICAL  
AND COMPUTER ENGINEERING  
PURDUE UNIVERSITY  
WEST LAFAYETTE, INDIANA 47907-1285

---

# Statistical Estimation of Combinational and Sequential CMOS Digital Circuit Activity Considering Uncertainty of Gate Delay Models

Tan-Li Chou and Kaushik Roy  
Electrical and Computer Engineering  
Purdue University  
West Lafayette, IN 47907-1285

Contact person: Kaushik Roy  
**Ph:** 317-494-2361  
**Fax:** 317-494-3371  
e-mail: [kaushik@ecn.purdue.edu](mailto:kaushik@ecn.purdue.edu)

# Statistical Estimation of Combinational and Sequential CMOS Digital Circuit Activity Considering Uncertainty of Gate Delay Models<sup>1</sup>

## Abstract

While estimating glitches or spurious transitions is challenge due to signal correlations, the random behavior of logic gate delays makes the estimation problem even more clifficult. In this paper, we present statistical estimation of signal activity at the internal and output nodes of **combinational** and sequential CMOS logic circuits considering uncertainty of gate delay models. The methodology is based on the stochastic models of logic **signals** and the probabilistic behavior of gate delays due to process variations, interconnect **parasitics**, etc. We propose a statistical technique of estimating average-case activity, which is flexible in adopting different delay models and variations and can be integrated with worst-case analysis into statistical logic design process. Experimental results show that the uncertainty of gate delay makes a **great** impact on activity at individual nodes (more than 100%) and total power dissipation as well.

---

<sup>1</sup>This research was supported in part by ARPA (under contract F33615-95-C-1625), by NSF CAREER award, and by IBM Corporation

# 1 Introduction

As mobile and portable information systems are becoming more popular, battery and packaging technologies do not seem to be keeping up with the same pace. On the other hand, deep submicron processes are pushing higher levels of integration, which increases the number of transistors: in VLSI circuits, and hence, the power density. Reliability problems such as run time **errors** due to overheating have become more and more serious. As a result, it has become important to consider power dissipation and reliability issues during the design phase. In order to design circuits for low power and high reliability, accurate estimation of power **dissipation** is required. This paper considers accurate estimation of dynamic power dissipation for **CMOS** digital circuits considering uncertainty in gate delay models. In CMOS digital circuits **majority** of the power dissipation is due to charging and **discharging** of load capacitances of logic gates. Such charging and discharging occur due to signal transitions, which depend on input signal patterns. Therefore, accurate estimation of power dissipation involves accurate estimation of signal switching activity at the internal nodes of a circuit. However, it is computationally too expensive to try all possible combinations of inputs in order to estimate power dissipation. Therefore, techniques are being developed to accurately estimate power dissipation considering probabilistic and statistical **approaches**. These techniques are **referred** to as weakly input-pattern dependent [7].

In the probabilistic approaches, the signals are usually modeled as stochastic processes having signal probability (probability of having a logic ONE) and activity (average number of signal switching per unit time). A single symbolic simulation run determines the internal node signal probability and activity by propagating the input probability information through the circuit [1, 2, 6]. For accuracy, the probabilistic approaches need to consider both spatial and temporal correlation of signals [2, 14, 15]. However, when it comes to sequential circuits, spatial and temporal correlations among state bits are more complicated, and were addressed in [8, 9, 4]. On the other hand, the statistical approaches simulate the circuits with a limited number of inputs vectors. The input vectors are randomly generated conforming to the given **signal** probability and activity of the input signals. The number of simulations for combinational circuits are determined by user-specified parameters, such as confidence levels and errors that can be tolerated [10, 11]. When sequential circuits are considered, the feedback **effect** of state bits and the near-close set problems must be dealt with [16, 3]. Readers are referred to a survey paper [7] for more details on some of the above techniques.

Some of the above techniques considered spurious transitions (glitches or **hazards**) while

others ignored. Spurious transitions appear in circuits not necessarily due to **design** errors. They also **depend** on the timing relationship between logic signals of the circuits. A node with inputs **having** different path delays in a synchronous circuit may have several transitions before it reaches steady state logic value. Though glitches can be reduced by balancing paths and by reducing the depth of circuits, the power dissipation caused by glitches can be as high as **70%** of the total power in some circuits such as combinational adders [7, 5]. The probabilistic techniques proposed to consider glitches while handling spatial and temporal correlations require building **BDDs** (Binary Decision Diagrams) of various forms [13, 6, 7], which can be computationally expensive. One of the problems that these techniques have is that inertial **delay** is not considered. As will be shown later, they usually overestimate the activity. **Moreover**, the activity at a node can be very sensitive to the path **delays** of its inputs as **pointed** out in [17]. Therefore, slight variation of delays result in great changes in activity. As a result, the minor delay model inaccuracies in the statistical and probabilistic approaches [6, 10, 11, 16, 3] can lead to large errors in estimated activity. Unfortunately, these minor inaccuracies are common due to sources of uncertainty such as process variations (die to die, wafer to **wafer**, and lot to lot), interconnect parasitics, temperature, approximation made by modeling gate delays, and other effects.

A solution has been proposed to estimate an upper bound on activity of individual nodes in larger circuits [17]. This technique uses signal uncertainty to capture the worst cases—to estimate the **maximum** activity. The technique provides information that can be used in worst case **design** techniques and is indeed very fast. However, worst case modeling can be extremely conservative and therefore can over-estimate the impact of **inaccuracies** in delay models on estimating glitches. In statistical designs of complex logic circuits for performance, a **methodology** proposed in [18] employs two stages of worst case analysis, calibrated with statistical simulation. The two stages of worst case analysis serve as filters to screen out circuits that easily meet their performance requirements. Similarly, statistical designs for low-power (low-activity) can take advantage of improved accuracy resulting from statistical **simulation**. In this paper, we propose a statistical estimation of CMOS circuit activity considering uncertainty of gate delay models. The **average-case** estimation together with the worst-case analysis can be used in statistical designs for low-power. For example, a two-stage design process is shown in Figure 1. The sources of uncertainty are represented by probability distributions in the statistical estimation. The first stage uses worst-case analysis to identify nodes that can have very high activity, which may result in reliability problems [1].

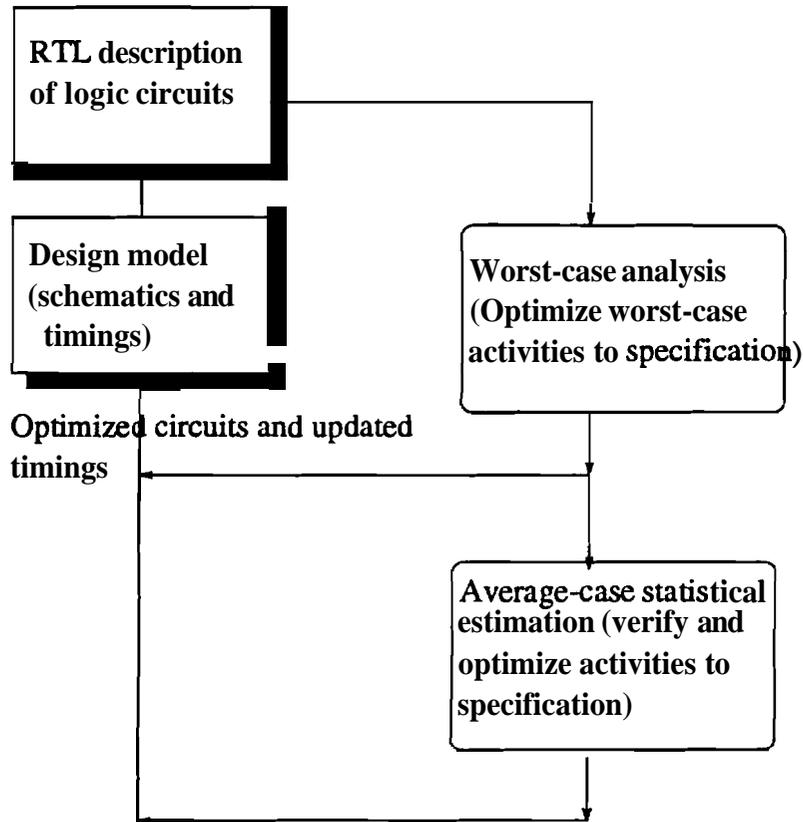


Figure 1: Design flow for statistical low-power design of logic circuits

Then one can reduce the activity (especially glitches) of these nodes by balancing paths (through transistor resizing, re-synthesizing the circuits, inserting buffers and latches, etc.). Circuit nodes that could not easily meet the power dissipation goal at the first stage will be passed down to the average-case statistical estimation stage for further analysis and optimization. In the second stage, average-case statistical estimation is performed providing more information than worst-case analysis. Therefore, one can utilize the more accurate information to optimize the nodes that failed the specification in the first stage. In addition, since in the second stage the average-case activity is known, one can further optimize the circuits for lower overall power dissipation.

The paper is organized as follows. Section 2 reviews signal probability and activity definitions and the relationship between activity and power dissipation. Monte Carlo methods, which are the basis of the proposed statistical estimation, will be introduced in Section 3. Section 3 also analyzes the errors in sequential circuit activity estimation when "near closed sets" are present in sequential circuits. Section 4 introduces the probabilistic model of gate delays and determines a statistical estimation method of how to take the random behavior of

gate delay into consideration. Experimental results are presented in section 5. Conclusions are given in Section 6.

## 2 Preliminaries and Definitions

This section formally defines signal probability and activity. A brief discussion on power dissipation in CMOS circuits is also presented.

### 2.1 Signal Probability and Activity

Given a logic signal  $x(t)$  and a random variable  $\tau$ , the companion process of  $x(t)$  is defined as  $\mathbf{x}(t) = x(t + \tau)$ , where  $\tau$  is uniformly distributed over  $\mathfrak{R}$  (real number). The **bold font** is used to represent a stochastic process. The primary inputs to a circuit are modeled as mutually independent companion processes of logic signals. It can be proved [1] that the probability of a companion process of a logic signal  $x(t)$  assuming the logic value ONE at any given time  $t$  ( $\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} \mathbf{x}(t) dt$ ) becomes a time constant and is called the equilibrium probability. This is denoted by  $P(x)$ . In contrast, the signal probability is defined as (clock cycles in which the signal is steady state ONE)/(total clock cycles). Note that steady state signals are only considered in signal probability estimation and any spurious transitions are ignored. Najm [1] has also shown that the activity  $A(x)$ , defined as  $\lim_{T \rightarrow \infty} \frac{n_x(T)}{T}$ , is equal to the expected value of  $\frac{n_x(T)}{T}$  (mean-ergodic). The variable  $n_x$  is the number of switching of  $x(t)$  in the time interval  $(-T/2, T/2]$ .

If we assume all primary inputs to the circuits under consideration switch only at the leading edge of the clock and the circuits are delay-free, we can define normalized activity. Normalized activity, denoted by  $a(x)$  is defined as  $A(x)/f$ , where  $A(x)$  and  $f$  are the activity at node  $x$  and clock frequency, respectively. Normalized activity has an intuitive meaning. That is, the probability of node  $x$  switching within a clock cycle. In circuits with arbitrary gate delays where glitches (or hazards) exist, we still define normalized activity  $a(x)$  as  $A(x)/f$ . However, note that  $a(x)$  can be greater than one. Hence,  $a(x)$  represents the average number of transitions in a clock cycle.

### 2.2 Power Dissipation in CMOS Logic Circuits

Of the three sources of power dissipation in digital CMOS circuits – switching, direct-path short circuit current, and leakage current – the first one is by far the dominant. Ignoring

power dissipation due to direct-path short circuit current and leakage current, the average power dissipation in a CMOS logic is given by  $POWER_{avg} = \frac{1}{2}V_{dd}^2 \sum_i C_i A(i)$ , where  $V_{dd}$  is the supply voltage,  $A(i)$  is the activity at node  $i$ , and  $C_i$  is the capacitive load at that node. The summation is taken over all nodes of the logic circuit. It should be observed that  $A(i)$  is proportional to  $a(i)$ .  $C_i$  is approximately proportional to the fanout at that node. As a result, the normalized power dissipation measure  $\Phi$  defined as  $\Phi = \sum_i fanout_i \times a(i)$  is proportional to the average power dissipation in CMOS circuits. The parameter,  $fanout_i$  is the number of fanouts at node  $i$ .

### 3 Monte Carlo Based Power Estimation for Sequential Circuits

In this section we will first give a short overview of the Monte Carlo techniques in estimation of signal activity followed by a detailed analysis of the errors introduced in estimation of circuit activity for sequential circuits when "near closed sets of states" are present. In the presence of such states, we also derive a technique to estimate power dissipation in sequential circuits.

The basic idea of Monte Carlo methods for estimating activity of individual nodes is to simulate a circuit by applying random pattern inputs. The convergence of simulation can be obtained when the activities of individual nodes satisfy some stopping criteria. The procedure is outlined in Figure 2.

We can use random number generators to generate input patterns conforming to the given probabilities and activities. During a given period, say  $T$  ( $T$  clock cycles), we count the number of transitions at each node,  $n_1$  and call the value  $n_1/T$  a random sample.  $T$  is called the sample length in this paper. The process is repeated  $K$  times to have  $K$  independent samples,  $a_j = n_j/T$ ,  $j = 1 \dots K$ , by using different seeds for the random number generators. The sample mean is defined as  $\bar{a} = (\sum_{j=1}^K a_j)/K$ . For large  $K$ ,  $\bar{a}$  will approach the expected value of  $\bar{a}$ , which is  $\lim_{T \rightarrow \infty} n_T/T$ , and is denoted as  $a$  since the signal at each node is mean-ergodic (section 2).  $n_T$  is the number of transitions in the time interval  $(\frac{-T}{2}, \frac{T}{2}]$ . Similarly, for large  $K$  the sample standard deviation  $s$  will approach the true standard deviation  $\sigma$ . Furthermore, according to the Central Limit Theorem [19]  $\bar{a}$  is a random variable with mean  $a$  and has a distribution approaching the normal distribution if  $K$  is large (typically  $\geq 30$ ). Likewise  $\sigma \approx s/\sqrt{K}$ . It has been shown in [10, 11] that for

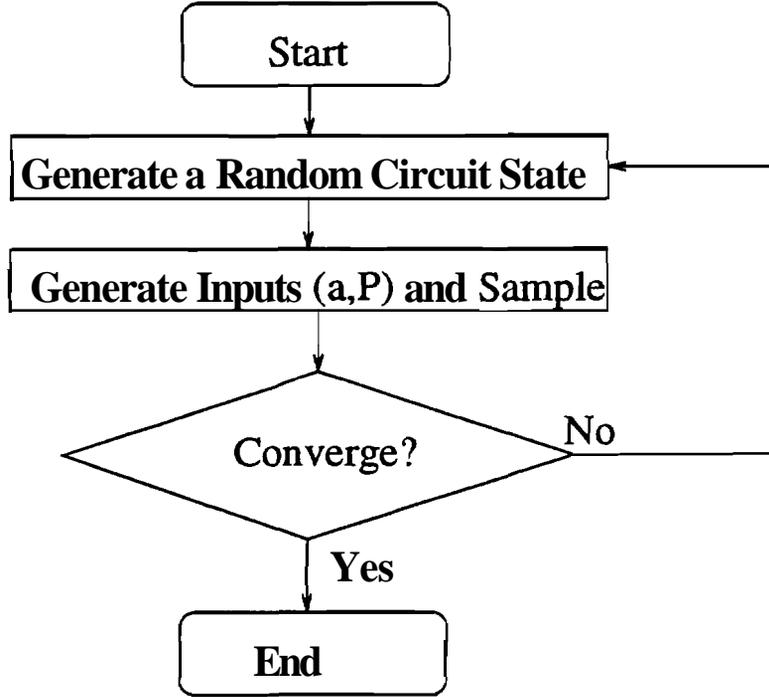


Figure 2: Monte Carlo based technique flow chart

$(1 - \alpha) \times 100\%$  confidence the following inequality holds:

$$\frac{|a - \bar{a}|}{\bar{a}} \leq \frac{z_{\alpha/2} s}{\bar{a} \sqrt{K}}, \quad (1)$$

where  $z_{\alpha/2}$  is a specific value such that the area under the standard normal distribution from  $z_{\alpha/2}$  to  $\infty$  is  $\alpha/2$ . Therefore, if

$$K \geq \left( \frac{z_{\alpha/2} s}{\bar{a} \epsilon'} \right)^2, \quad (2)$$

we have  $\frac{|a - \bar{a}|}{\bar{a}} \leq \frac{z_{\alpha/2} s}{\bar{a} \sqrt{K}} \leq \epsilon'$ , and hence  $\frac{|a - \bar{a}|}{a} \leq \frac{\epsilon'}{1 - \epsilon'} = \epsilon$ .

Equation 2 is the stopping criterion for  $(1 - \alpha) \times 100\%$  confidence and  $\epsilon$  is an upper bound on the relative error.

If any node in the circuit has a very low activity, that is, its  $a \ll 1$ , by equation 2 the number of samples required can be very large. This results in slow convergence. However, since these low-activity nodes contribute little to power dissipation, a modified stopping criterion is proposed in [11]. One can specify a particular threshold value  $a_{min}$  below which the activities of nodes are less important. Hence one may not wait for those nodes to converge to a value within a certain percentage of error. Furthermore, if

$$K \geq \left( \frac{z_{\alpha/2} s}{a_{min} \epsilon'} \right)^2, \quad (3)$$

we have  $\frac{|a - \bar{a}|}{\bar{a}} \leq \frac{z_{\alpha/2} s}{\bar{a} \sqrt{K}} \leq \frac{a_{min} \epsilon'}{a'}$  and hence  $|a - \bar{a}| \leq a_{min} \epsilon'$ .

Therefore, equation 3 becomes the stopping criterion (with  $\bar{a} < a_m$ ) for  $(1 - \mathbf{a}) \times 100\%$  confidence and  $a_{min} \epsilon'$  is an absolute error bound (not a percentage error bound)..

In sequential circuits, things are different due to the state-bit feedback. One of the approaches is to monitor the state-bit probabilities to determine the convergence [16]. In order to have  $(1 - \mathbf{a}) \times 100\%$  confidence and some error  $\epsilon$  (upper bound on absolute error of the state-bit probabilities), one must perform at least  $K \geq \max(N_1^2, N_2^2, N_3^2)$  runs, where:

$$N_1 = \frac{z_{\alpha/2}}{2\epsilon}, \quad N_2 = \frac{z_{\alpha/2} \sqrt{2\epsilon + 0.1} + \sqrt{(\epsilon + 0.1) z_{\alpha/2}^2 + 3\epsilon}}{2\epsilon}, \quad N_3 = \frac{\sqrt{63} + z_{\alpha/2}}{2\sqrt{\epsilon}}. \quad (4)$$

However, to derive each sample probability of a state bit is a problem since the probability depends on the state the sequential circuit is in. This can be resolved as follows. Assume that the state of the machine at time  $k$  ( $k^{\text{th}}$  clock cycle) becomes independent of its initial state at time 0 as  $k \rightarrow \infty$ . As a result, the probability that the state bit signal  $\mathbf{s}_i(k)$  is logic **ONE** at time  $k$  with initial state  $S(0)$  being  $S_0$ , denoted as  $P(\mathbf{s}_i(k) = 1 | S(0) = S_0)$  (abbreviated as  $P_k(\mathbf{s}_i | S_0)$ ), has the following property:

$$\lim_{k \rightarrow \infty} P_k(\mathbf{s}_i | S_0) = \lim_{k \rightarrow \infty} P_k(\mathbf{s}_i = 1) = P(s_i). \quad \text{Similarly,} \quad (5)$$

$$\lim_{k \rightarrow \infty} P_k(\mathbf{s}_i | S_1) = P(s_i). \quad (6)$$

That is, the probability will be independent of the initial state as  $k \rightarrow \infty$ . Based on this property, two runs starting from two different initial states  $S_0$  and  $S_1$  are performed at the same time. During the simulation, the difference and average of  $P_k(\mathbf{s}_i | S_0)$  and  $P_k(\mathbf{s}_i | S_1)$  are monitored. If both the difference and average remain within a window of width of  $\epsilon_w$  for a certain user-specified number of consecutive clock cycles, it is declared that  $P_k(\mathbf{s}_i | S_0)$  and  $P_k(\mathbf{s}_i | S_1)$  have converged. This convergent value of probability is a sample value. The sample procedure is repeated  $K$  times to meet the user-specified error and confidence level, as mentioned earlier in this paragraph. However, there are some circuits that can not be directly estimated by this approach.

Let us consider the State Transition Graph (STG) of Figure 3. Let  $G_1$  and  $G_2$  denote the set of states given by  $\{s1s0, \bar{s}1\bar{s}0\}$  and  $\{\bar{s}1s0, \bar{s}1s0\}$ , respectively. Assumt: that the probability of making a transition between the set of states in  $G_1$  and the set of states in  $G_2$  is very low. As a result, most of the samples are collected from  $G_1$  if the initial states are  $s1s0$  ( $S_0$ ) or  $\bar{s}1\bar{s}0$  ( $S_1$ ). These sets  $G_1$  and  $G_2$  are called sets of near-close states.

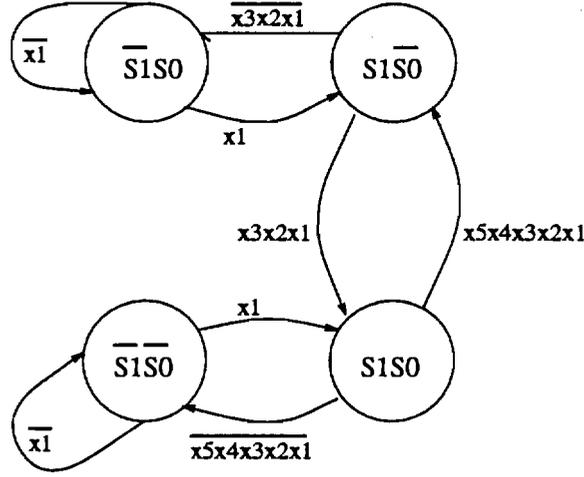


Figure 3: Example 2: Another STG of a sequential logic circuit.

Let  $y$  be the output node given by  $y = (s1s0 + \bar{s}1\bar{s}0)x1$ . Considering only the set of states in  $G_1$ ,  $y = (s1s0 + \bar{s}1\bar{s}0)x1 = x1$ , while considering only the set of states in  $G_2$ ,  $y = (s1s0 + \bar{s}1\bar{s}0)x1 = 0$ . Therefore,  $P(y) = P(x1)$  in  $G_1$  and  $P(y) = 0$  in  $G_2$ . That is, the probability behavior is very different in two different groups of states. Data sampled from a particular group is biased giving errors. As a matter of fact, if we assume all the primary inputs have the same probability and normalized activity of 0.2 and 0.3 respectively, the normalized activity of  $y$  sampled from  $G_1$  is 0.3 ( $a(y, G_1)$ ) and 0.02 ( $a(y, G_2)$ ) from  $G_2$ .

A solution to this problem is as follows. Let us assume that we know the values of  $P(G_1)$  and  $P(G_2)$  of the previous example. The normalized activity,  $a(y)$  can be computed as follows,

$$a(y) = P(G_1) \times a(y, G_1) + P(G_2) \times a(y, G_2). \quad (7)$$

If STG is given,  $P(G_1)$  and  $P(G_2)$  may be computed by assuming that the primary inputs are either temporally uncorrelated [8, 9] (which may give errors when they are not) or Markov [4]. A primary input is Markov if its future value depends on the present value and does not depend on its past. However, if no STG knowledge is assumed, can we find out  $P(G_1)$  and  $P(G_2)$ ? Under the assumption that the primary inputs are Markov, it turns out that we can implicitly compute  $P(G_1)$  and  $P(G_2)$ . Based on the assumption that the primary inputs are Markov, a new state transition graph called Extended STG (ESTG) is built by transforming the original STG according to some rules. The resultant ESTG (rather than STG) is Markov [4]. Therefore, there is a transition matrix that corresponds to ESTG. It is found that  $P_{warmup}^k(G_i)$ , the probability of reaching one of the states of  $G_i$  at the end

of  $k$  clock cycles ( $k \geq$  a certain period of time called the **warmup** period) with any initial state is very close to  $P(G_i)$ . The error specified by the users determines the **warmup** period according to the following empirical inequality that we derived in [3],

$$|P_{\text{warmup}}^k(G_i) - P(G_i)| \leq N_s |\lambda_2|^k, \quad (8)$$

where  $\lambda_2$  is the second largest eigenvalue (absolute value) of the transition matrix and  $N_s$  is the number of ESTG states. The upper bound on the number of states of the ESTG is  $2^{i+j}$ , where  $i$  and  $j$  are the number of state bits and the number of inputs, respectively. Therefore, if we specify the upper bound on the relative error  $\epsilon_G$  that  $P(G_i)$  can have, we have

$$k \geq \frac{\ln \frac{N_s}{\epsilon_G P(G_i)}}{\ln 1/|\lambda_2|}. \quad (9)$$

Equation 8 implies that if we starting with a randomly generated initial state simulate the circuit for a **warmup** period of clock cycles and then sample data, the probability of sampling data from among the states of  $G_i$  is  $P(G_i)$ . If we repeat the same procedure  $N$  times, we will have  $N \cdot P(G_1)$  samples from  $G_1$  and  $N \cdot P(G_2)$  samples from  $G_2$ . Let  $a_j(y|G_i)$  represent the  $j^{\text{th}}$  sample taken from  $G_i$ . Hence the mean of the samples taken from  $G_i$  is  $\frac{\sum_{j=1}^{N \cdot P(G_i)} a_j(y|G_i)}{N \cdot P(G_i)}$ , denoted as  $a(y|G_i)$ . As a result, the mean of these samples is

$$\begin{aligned} \frac{(\sum_{j=1}^{N \cdot P(G_1)} a_j(y|G_1) + \sum_{j=1}^{N \cdot P(G_2)} a_j(y|G_2))}{N} &= \frac{N \cdot P(G_1) a(y|G_1) + N \cdot P(G_2) a(y|G_2)}{N} \\ &= P(G_1) a(y|G_1) + P(G_2) a(y|G_2), \end{aligned}$$

which is not biased. However, since STG is not given, ESTG and its corresponding transition matrix and hence  $\lambda_2$  and  $N_s$  can not be derived. To be conservative, we may choose  $\lambda_2$  to be 0.9 and  $N_s$  to be  $2^r$  ( $r$  is the number of primary inputs and state bits), which is an upper bound on the number of ESTG states. The modified version of Monte Carlo based technique for sequential circuits is outlined in Figure 4. It is worth mentioning is that the **warmup period simulation** does not need any delay model. What matters is the steady state logic value (it may have some spurious transitions) of the state bits rather than the transient behavior.

Lastly, there is only one problem left. If we examine the assumption of the stopping criterion (Inequality 2), it is assumed that the mean  $a$  has a normal distribution. But in the previous example, apparently  $a(y)$  has a bimodal distribution. Can we apply the same stopping criterion to the bimodal distribution in this case? In order to answer this question, we will compare the user-specified relative error ( $\epsilon'$  in Inequality 2) with the actual

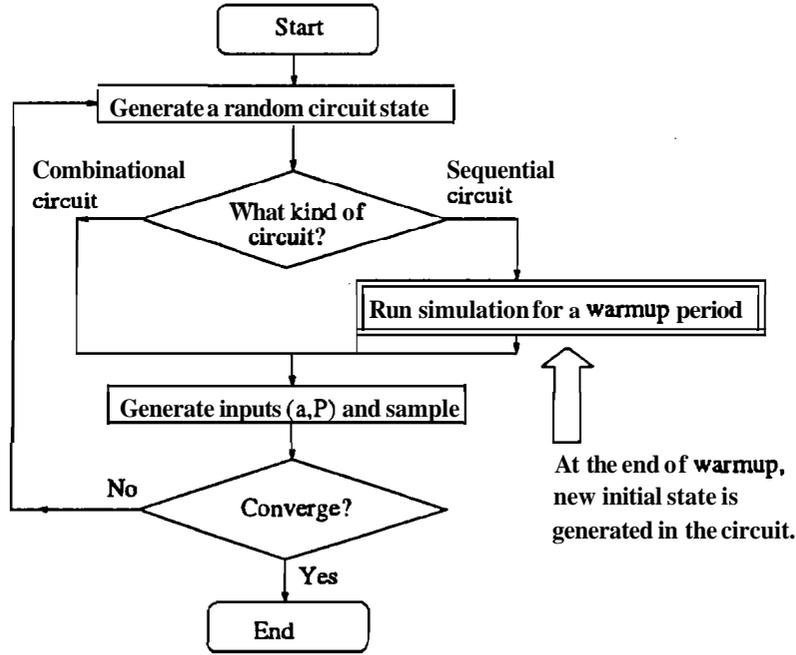


Figure 4: Monte Carlo based technique flow chart for both sequential and combinational circuits

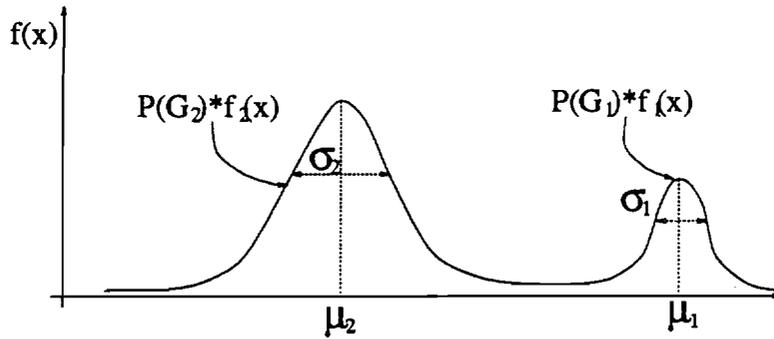


Figure 5: Birnodal distribution as a linear combination of two normal distributions

relative error ( $\epsilon_b$ ) resulting from applying the stopping criterion (Inequality 2) to a bimodal distribution. Let us assume that the bimodal distribution function is a linear combination of two normal distribution functions  $f_1(x)$  and  $f_2(x)$ . That is,

$$f(x) = P(G_1) \cdot f_1(x) + P(G_2) \cdot f_2(x), \quad (10)$$

where  $P(G_1)$  and  $P(G_2)$  are steady-state probabilities of two groups of states as mentioned earlier (two near-close sets) and  $P(G_1) + P(G_2) = 1$ .  $f_i(\cdot)$  represents the population of  $G_i$ . It is justified by assuming  $T \geq 30$  that  $f_i$  is normal, where  $T$  is the number of clock cycles in each sample. It is illustrated pictorially in Figure 5

Assume that in each sample we start to take data after simulating the sequential circuit

for a **warmup** period. If a total number of  $N$  samples are taken when the stopping criterion is met, we have  $N_i$  ( $\approx N \cdot P(G_i)$ ) samples collected from  $G_i$ .  $N$  is determined by the stopping criterion (Inequality 2) to meet the user-specified relative error  $\epsilon$  and  $(1 - \alpha) \times 100\%$  confidence level. However, it is based on the assumption that the distribution is normal **rather** than bimodal. Without loss of generality, let us assume  $P(G_1) \leq P(G_2)$ . If we use  $N_i$  samples to estimate  $\mu_i$ , we have  $s_i$  and  $m_i$  as sample standard deviation and sample mean. It can be shown (see Appendix A) that  $\frac{\epsilon_b}{\epsilon'}$ , the ratio of the resultant relative error to the user-specified relative error, with the same confidence level is,

$$\frac{\epsilon_b}{\epsilon'} \leq \frac{\left( \sqrt{\frac{N_1}{N}} r_m r_{rsm} t_{\alpha/2} + \sqrt{\frac{N_2}{N}} z_{\alpha/2} \right) \sqrt{N-1}}{z_{\alpha/2} \sqrt{r_m^2 r_{rsm}^2 (N_1 - 1) + (N_2 - 1)}} \quad (11)$$

where  $r_m = \frac{m_1}{m_2}$ ,  $r_{rsm} = \frac{s_1/m_1}{s_2/m_2}$ , and  $t_{\alpha/2}$  and  $z_{\alpha/2}$  are obtained from the *t-distribution* and normal distribution [20]. A few plots are shown in Figures 6 through 9. Surprisingly the ratios are less than 1.5 for most of the values of  $P(G_1)$  with different parameters. This implies that in order to ensure the actual relative error to be less than  $\epsilon_b$ ,  $\epsilon'$  in the stopping criterion (Inequality 2) has to be less than  $\frac{\epsilon_b}{1.5}$ . For example, if we assume 7.5% error to be tolerable ( $\epsilon_b = 0.075$ ), then  $\epsilon'$  must be less than  $\frac{7.5}{1.5} = 5\%$  ( $\epsilon' = 0.05$ ). The worst case only occurs when  $P(G_1)$  is very small (that is, when only a couple of samples are collected from  $G_1$ ). For example,  $\frac{\epsilon_b}{\epsilon'}$  is equal to 1.52 as  $P(G_1) = 0.08$ ,  $r_m = 10$ , and  $r_{rsm} = 0.5$ . Several other observations can also be made. The larger the  $r_m$  is ( $m_1$  is greater than  $m_2$ ), the higher ratio of relative error is when  $P(G_1)$  is very small (less than 0.05). This is shown in Figure 6 and can be explained as follows. Since only a few samples are collected from  $G_1$  while  $m_1$  is much larger than  $m_2$ , we may expect higher error ratio when the ratio  $r_m$  ( $\frac{m_1}{m_2}$ ) is larger. On the other hand, when  $m_1$  is smaller than  $m_2$  (Figure 7) and even when only a couple of samples are from  $G_1$ , it does not really effect the error since  $P(G_1)$  is very small. It is also observed (Figure 8) that with more samples (greater  $N$ ) the error ratio is smaller when  $P(G_1)$  is small. Another factor that affects the error ratio is  $r_{rsm}$ , which is the ratio of relative sample standard deviation  $\frac{s_1}{m_1}$  to  $\frac{s_2}{m_2}$ . When only a couple of samples are taken from  $G_1$  and they have greater relative sample standard deviation, it is also expected that the ratio will be higher, which is shown in Figure 9.

In the above analysis, we have assumed that there are only two near-close sets. But there can be more than two near-close sets in a sequential circuit. Our technique can be extended to cases having multiple near-close sets. In addition, the experimental results (Section /refresult) show that accurate results can be obtained for the sequential ISCAS

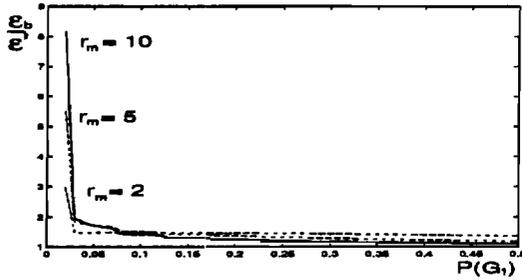


Figure 6: Relative error ratio with  $r_{rsm} = 1$  and  $N = 120$ .

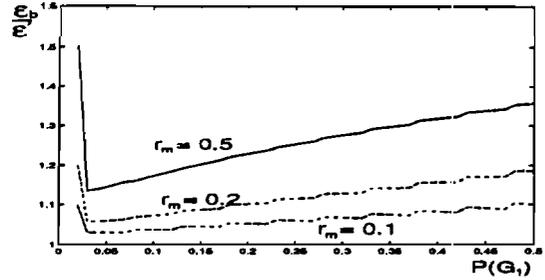


Figure 7: Relative error ratio with  $r_{rsm} = 1$  and  $N = 120$ .

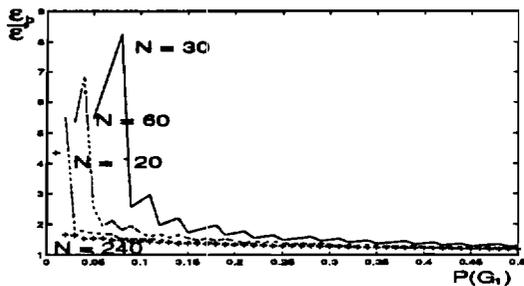


Figure 8: Relative error ratio with  $r_{rsm} = 1$  and  $r_m = 5$ .

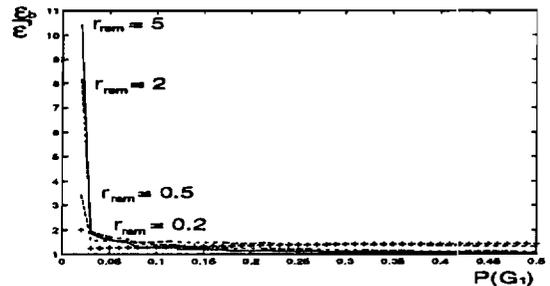


Figure 9: Relative error ratio with  $r_m = 5$  and  $N = 120$ .

benchmark circuits under this assumption.

## 4 Delay Models and Statistical Estimation

As mentioned earlier, minor delay model inaccuracies may lead to large errors in estimated activity. Therefore, delay models are crucial to the statistical estimation of activity. Probabilistic delay models used in the estimation will be introduced to capture the uncertainty of gate delays. Based on the probabilistic delay models, we will generalize the Monte Carlo approach.

### 4.1 Delay models

In the design phase, a designer is faced with different sources of uncertainty that affect the delays of the circuit. These sources can be grouped into two classes: **systematic** and **random** [18]. The systematic class includes approximations made to simplify the model for improving simulation time, approximations made to estimate device and interconnect **parasitics** prior to layout, and uncertainty in the final process center and distribution when

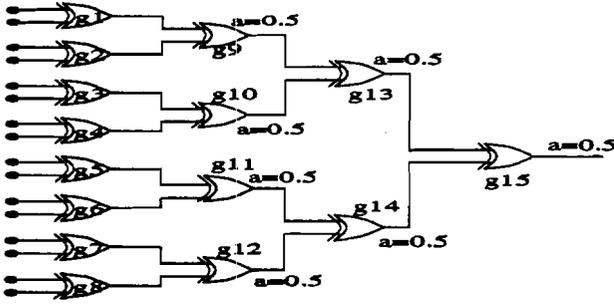


Figure 10: A Circuit with nominal delays

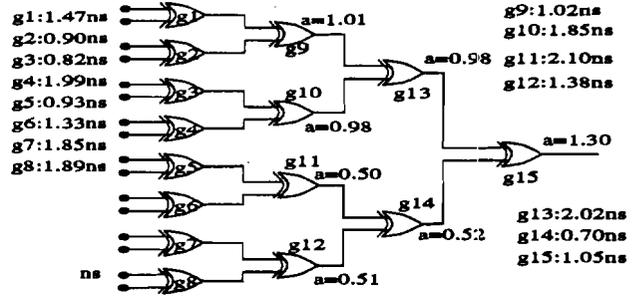


Figure 11: A Circuit with random delays

design proceeds in parallel with process development. On the other hand, the **random** class includes uncontrolled variations in photolithography, die to die variations, wafer to wafer variations, lot to lot variations, operating temperature, power supply voltage, etc. In [17], it has been shown that a circuit node where two reconvergent paths with different delays meet may have a large number of spurious transitions. However, even in a tree-structured circuit with balanced paths (without reconvergent fanout) there can be a large number of spurious transitions due to slight variations in delays. These variations can be caused by any of the above sources of uncertainty.

Let us consider the circuit of Figure 10. All gates are assumed to have the same delay. Because the tree has perfectly balanced paths, there are no glitches at all. The final output has normalized activity 0.5 when all the primary inputs are assumed to be synchronous and have activity of 0.5. However, due to sources of uncertainty, the gate delays may have variations and are shown in Figure 11. As a result, glitches do occur and the values of activities at individual nodes change. This is shown in Figure 11. The inertial delays are assumed to be half of the values of transport delays for the simulation. Notice that the final output normalized activity becomes 1.30 rather than 0.5. In order to capture this random behavior in statistical design, these sources of uncertainty are represented by probability distribution while in worst-case design, the extreme cases are taken into account.

In this paper, we choose transport delay ( $d$ ) model with inertial delay ( $d_I$ ). However, it should be noted that the technique is not restricted to such a delay model. The point is to model the parameters of chosen delay models as random variables in order to capture the probabilistic behavior of gate delays. The transport delay is modeled as a random variable of truncated normal distribution with mean  $\mu_d$  and standard deviation  $\sigma_d$  as shown in Figure 12. The mean is the nominal value of transport delay  $d$  and the deviation is either assigned by users or determined by feedback from the fabricated chips. Moreover, if a random delay is

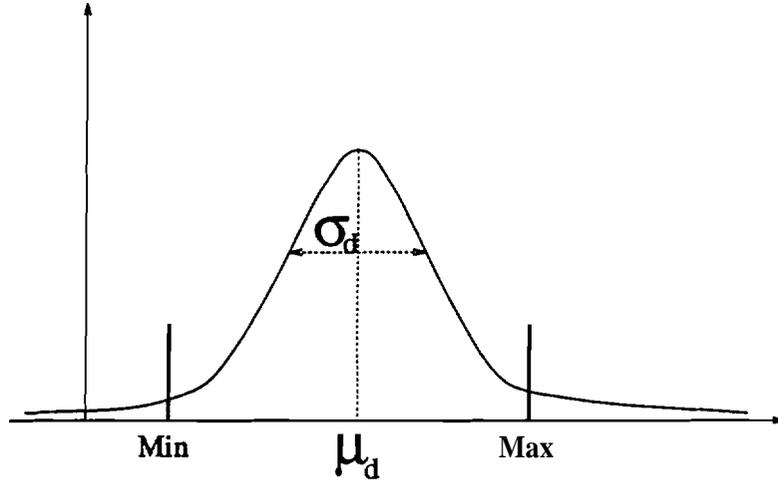


Figure 12: Random delay with truncated normal distribution

less than a **minimum** value **Min**, it is discarded since in real circuits it must be larger than some positive value. Similarly, if a random delay is greater than a **maximum value Max**, it is truncated since it can be considered as a delay fault.

## 4.2 Statistical Estimation

Recall that in Monte Carlo based technique the primary input patterns are generated conforming to a given activity and probability of the input signals. In a more abstract view point, we can think of activity ( $a$ ) at a node as a function of primary input vectors  $\mathbf{PI}$ . Each component of  $\mathbf{PI}$  is a stochastic process (see Section 2). Therefore,  $a$  is also a stochastic process and can be expressed as follows,

$$a = F(\mathbf{PI}). \quad (12)$$

In Section 3, we applied Monte Carlo based techniques to estimate the expected value of  $a$ ,  $E(a)$ . However, what is missing in this approach is the information about the delay. In other words, the delays of the gates of the circuit are assumed to be some constants (**deterministic**). Now assume that gate delays are not deterministic and each gate delay can be represented by a random variable  $d_i$ . If  $\mathbf{D}$  is a random vector consisting of all the random variables of gate delays,  $a$  can be represented as follows,

$$a = F(\mathbf{PI}, \mathbf{D}). \quad (13)$$

Therefore, when applying Monte Carlo based techniques to estimating  $F(\mathbf{PI}, \mathbf{D})$ , delays are modeled as random variables and should be generated from time to time along the

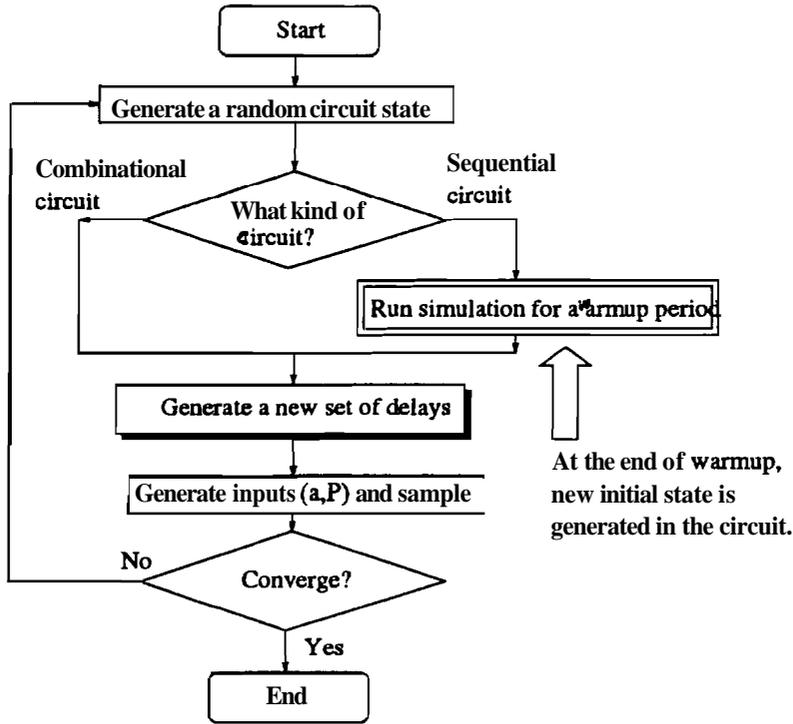


Figure 13: Modified Monte Carlo Based Technique Flow Chart

simulation. The rationale behind this is that whenever we generate a new set of delays, they correspond to another die or even the same die but with different operating conditions such as temperature and power supply voltage. In contrast to Figure 2, Figure 13 outlines the procedure of how the modified Monte Carlo based technique works.

## 5 Experimental Results

The Monte Carlo based approach with Probabilistic Delay models (MCPD) to estimate activities at the internal and output nodes of both combinational and sequential circuits has been implemented in C under the Berkeley SIS environment. In this section, in order to compare the results with gate delays to those assuming no delays, all the primary inputs are assumed to be synchronous. That is, if they switch, they switch at the same time (say, at the leading edge of a clock cycle). Primary inputs are randomly generated conforming to the given probability and activity of the input signals. In our analysis all the primary inputs are assumed to have signal probability of 0.5 and normalized activity of 0.5. MCPD uses the probabilistic delay model (Section 4). The transport delay  $d$ , which is a random variable, has mean  $\mu_d$  (equal to nominal value) and standard deviation  $\sigma_d$ . Unless mentioned otherwise, we assume that the standard deviation  $\sigma_d$  is equal to  $0.3\mu_d$  and the inertial delay

Table 1: Long run information and results of different delay models on ISCAS combinational benchmark circuits

Ckt	#gates	#levels	CPU	$\Phi_n$	$\Phi_l$	$\Phi_r$	$\Phi_m$	$\Phi_u$
C432	160	17	1.2 hr	129.5	197.3	196.4	193.7	214.8
C499	272	11	1.2 hr	184.2	276.4	275.7	270.2	269.8
C880	333	24	2.3 hr	278.9	394.7	394.1	389.4	390.9
C1355	546	24	5.1 hr	393.0	803.8	804.7	724.8	887.4
C1908	880	40	23.0 hr	625.1	1438.9	1446.2	1515.4	1634.8
C2670	1193	32	29.8 hr	892.4	1766.1	1769.0	1773.4	1720.0
C3540	1669	47	28.7 hr	1069.5	2568.8	2571.7	2576.6	2544.1
C5315	2307	49	51.9 hr	1984.4	4297.9	4306.2	4597.6	4605.6
C6288	2406	124	140.6 hr	2011.2	31200.8	31138.0	51502.7	57299.9
C7552	3512	43	56.2 hr	2691.2	6451.0	6432.4	6676.8	6903.1

is also a random variable that equals to  $0.5d$ . In order to assess the accuracy of the results, we run *MCPD* for a long time with 99% confidence and 1% error. For combinational circuits, since the activity is higher than that with zero delay models due to the presence of spurious transitions, we choose a higher threshold ( $a_{th} = 0.5$ ). In Table 1 the number of gates (#gates), maximum levels (#levels) and CPU time for long run *MCPD* (in hours on a SPARC 5 workstation) are provided. The same table also presents several power dissipation values (normalized power dissipation measure) of different delay models. Both  $\Phi_l$  and  $\Phi_r$  use the same probabilistic delay model but the former represents the long run results while the latter the normal run results. The normal run result is the result by assuming 95% confidence, 5% error, and threshold of  $a_{th} = 0.5$ .  $\Phi_n$ ,  $\Phi_m$  and  $\Phi_u$  represent the results of no delays, transport delay (not random but equal to nominal value) with inertial delay being half of the transport delay, and unit delay, respectively.

Several interesting observations can be made by examining at Table 1. Though the results of unit delay model can have higher spurious transitions (6 out of 10 circuits of Table 1), it is not always the highest. As far as  $\Phi_r$  and  $\Phi_m$  are concerned,  $\Phi_r$  is not necessarily greater than  $\Phi_m$ . It depends on types of circuits. For example, for the first four circuit;  $\Phi_r > \Phi_m$  while for the rest of the circuits  $\Phi_r < \Phi_m$ . That is, for some circuits like C499, activity at some nodes is sensitive to uncertainty and therefore  $a_{th}$  (activity with probabilistic delay) is greater than  $a_{th}$  (activity with non-random nominal delay). This is shown in Figure 14. Also note that the estimated activity at individual node with probabilistic delay models can be twice as high as that with nominal delay models. On the other hand, for some circuits like C6288, uncertainty helps balance paths of circuits and results in fewer spurious transitions

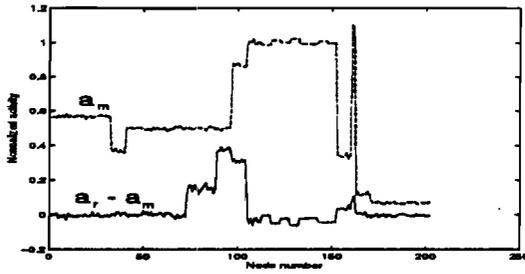


Figure 14: Activity sensitive to uncertainty (C499)

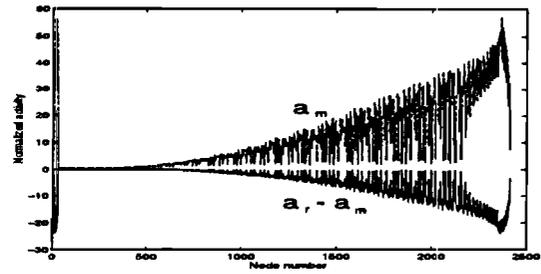


Figure 15: Uncertainty helping balance paths of circuits (C6288)

(Figure 15). The estimated activity at individual node with nominal delay models can be twice as high as that with probabilistic delay models. Another interesting observation is that when activity at a node is sensitive to uncertainty, the higher the uncertainty is, the more spurious transitions the node has. However, when uncertainty helps balance paths of circuits, the higher the uncertainty is, the less spurious transitions the node has. These are illustrated by Figure 16 (C499) and Figure 17 (C1908), where  $\Phi$  represents the normalized power dissipation measure and  $a$  and  $\mu$  are the standard deviation and mean of the transport delay.

The accuracy of the normal run results are shown on Table 2. The average relative error (% error) is the average percentage of all the relative errors of individual nodes provided that  $a(y) \geq 0.1$ . This indicates that on the average how accurate the long run MCPD is for those nodes with higher activity. Maximal absolute error (Max abs error) among all the nodes is also given. The term  $a$  represents the activity of a node at which maximal absolute error occurs and its percentage error is denoted as a % error in the table. Notice that, the CPU time of normal **run** MCPD is comparable to that of statistical techniques without considering probabilistic **behavior** of logic gate delays [11].

Similarly, for **sequential** circuits we run MCPD for a long time with 99% confidence, 1.5% error, 0.1 threshold ( $a_{i,j}$ ), 2nd largest eigenvalue ( $\lambda_2$ ) of 0.99, near-close set probability ( $P(G_i)$ ) of 0.1, and the upper bound on the relative error of  $P(G_i)$  ( $\epsilon_G$ ) being 1%. The results are shown in Table 3. Number of primary inputs (#PI) and latches (#ff), length of **warmup** period (**warmup** length), and CPU time in seconds on SPARC 5 workstations are also shown. In contrast to long run MCPD, the **normal** run MCPD takes the following parameters: 95% confidence, 7.5% error, 0.3 threshold ( $a_{i,j}$ ), 2nd largest eigenvalue ( $\lambda_2$ ) of 0.9 near-close set probability ( $P(G_i)$ ) of 0.5, with  $\epsilon_G$  being 5%. Comparison between results

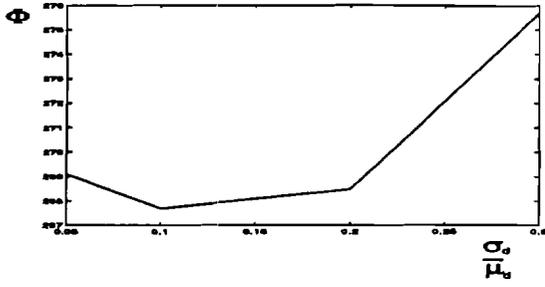


Figure 16: Activity sensitive to uncertainty (C499)

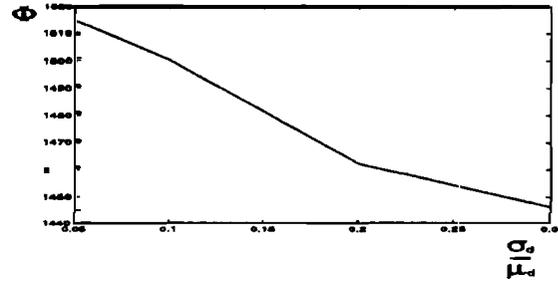


Figure 17: Uncertainty helping balance paths of circuits (C1908)

Table 2: Individual node information on *MCPD* in comparison with long run *MCPD* results

Ckt	CPU (sec)	Ave. abs error	% error	Max abs error	a	a % error
C432	89.2	0.0067	1.2	0.052	0.96	5.4
C499	78.3	0.0065	1.1	0.034	1.23	2.8
C880	110.9	0.0073	1.6	0.032	1.07	3.0
C1355	333.3	0.0062	1.3	0.029	2.05	1.4
C1908	807.5	0.0073	0.84	0.045	1.07	4.2
C2670	938.2	0.0067	0.87	0.038	1.47	2.5
C3540	2140.7	0.0060	1.05	0.047	4.60	1.0
C5315	3141.4	0.0072	0.86	0.087	3.60	2.4
C6288	10703	0.040	0.83	0.343	14.37	2.4
C7552	4988.7	0.0075	0.82	0.054	2.64	2.1

Table 3: Long run results on ISCAS sequential benchmark circuits

Ckt	#gates	#PI	#ff	#levels	warmup length	Sample length	#samples	CPU (sec)
s344	160	9	15	20	2343	2500	678	12834
s382	158	3	21	9	2343	2500	500	5115
s400	162	3	21	9	2343	2500	498	5064
s444	181	3	21	11	2343	2500	498	4928
s526	193	3	21	9	2343	2500	353	3359
s641	379	35	19	74	4412	4500	459	29986
s713	393	35	19	74	4412	4500	308	21204
s832	287	18	5	10	2274	2500	760	20393
s953	418	16	29	16	3791	4000	682	26426
s1196	529	14	18	24	2894	3000	704	52718
s1238	508	14	18	22	2894	3000	674	48657
s1423	657	17	74	59	6963	7000	395	85709

Table 4: Individual node information on *MCPD* in comparison with long run *MCPD* results

Ckt	# samples	CPU (sec)	NDQ	PDQ	Ave. abs error	% error	Max abs error	$a$	$a$ % error
s344	146	119.3	83.8	106.0	0.0033	0.99	0.013	0.32	3.9
s382	93	41.8	48.5	60.2	0.0023	1.8	0.014	0.25	5.34
s400	97	27.8	49.0	57.3	0.0113	8.9	0.17	0.83	20.9
s444	75	23.0	53.2	63.8	0.0075	6.0	0.11	0.77	14.2
s526	50	16.4	62.4	72.2	0.0078	7.4	0.29	0.62	47.6
s526a	90	44.3	62.4	78.5	0.0014	1.1	0.010	0.27	3.82
s526b	120	58.6	62.4	77.8	0.0009	0.7	0.009	0.44	1.94
s641	88	225.8	155.5	191.3	0.0041	1.4	0.019	0.54	3.6
s713	95	250.0	170.1	214.5	0.0034	1.2	0.015	0.29	5.0
s832	99	137.5	186.5	228.1	0.0029	1.3	0.012	0.75	1.6
s953	162	260.0	140.3	162.2	0.0031	2.1	0.015	0.30	4.9
s1196	149	515.1	334.6	420.2	0.0036	1.1	0.016	1.10	1.4
s1238	134	455.4	345.0	434.4	0.0043	1.3	0.023	0.62	3.8
s1423	126	605.9	290.1	418.2	0.0064	2.2	0.039	1.26	3.0

Table 5: Individual node information on s526 *MCPD* with different number of samples

N	1	2	3	4	5	6	7	8	9	10	11	12
50	47.6	47.6	64.1	64.1	14.0	14.0	45.7	45.7	17.3	8.1	8.6	8.4
90	0.87	0.87	3.28	3.28	1.05	1.05	1.08	1.08	0.63	3.82	0.42	0.03
120	0.09	0.09	2.03	2.03	0.36	0.36	0.36	0.36	0.24	0.79	0.33	0.44

of long run *MCPD* and normal run *MCPD* is shown in Table 4. In the table, Ave. abs error, % error, **Max** abs error,  $a$  and  $a$  % error have the same meaning as those of Table 2. NDQ and PDQ represent the normalized power dissipation with zero delay and with probabilistic delay, **respectively**. Notice that we assume that there are only two near-close sets. Therefore, the activity **sample** can be taken from a bimodal population ( $G_1$  and  $G_2$ ) rather than a single population ( $G_1$  or  $G_2$ ). Though the stopping criterion (Inequality 2) is derived based on the assumption that the distribution is normal, we can still apply it with slight **modification**. This is shown **as** Follows. When a user specifies the upper bound on relative error to be  $\epsilon_b$ ,  $\epsilon'$  in Inequality 2 must be less than or equal to  $\frac{\epsilon_b}{1.5}$  (Section 3). For example, when the user specify relative error to be 7.5%, Inequality 2 will be applied with  $\epsilon'$  being 5% to check if the simulation **converges**.

Notice that the sequential circuits s400, s444, and s526 have higher relative error over

Table 6: Run time test for large sequential circuits

Ckt	#gates	#PI	#ff	#levels	CPU
s9234	5597	36	211	58	3.6 hr
s13207	8027	31	669	59	8.4 hr
s15850	9786	14	597	82	16.3 hr

the long run results. It is observed that these three circuits have smaller number of samples than 120. **Especially**, s526 has only 50 samples. Recall the discussion on bimodal population sampling in Section 3. The ratio  $\frac{\epsilon_r}{\epsilon_n}$  of relative error assuming bimodal population to relative error assuming normal distribution is higher when the number N of samples is smaller. This is shown in Figure 8. Therefore, we suspect that the high percentage error can be attributed to the smaller number of samples than necessary. That is, the ratio  $\frac{\epsilon_r}{\epsilon_n}$  is **greater** than 1.5. Therefore, we also tried cases of N being 90 and 120. The results are shown in the table as s526a (N=90) and s526b (N=120), respectively. In Table 5 we examine the activities at 12 different nodes that have relative error more than 7.5% when N=50. The 7.5% relative error is what we specified up-front. As the number of samples increase, the relative error drops dramatically. **This** can be explained from Figure 8 as follows. When N increase **from** 60 to 120, the ratio  $\frac{\epsilon_r}{\epsilon_n}$  **drops** dramatically, too. Therefore, the 1.5 ratio assumption for  $\frac{\epsilon_r}{\epsilon_n}$  is met and the relative error falls within the user-specified range.

In order to test the run time for larger sequential circuits, three larger circuits (s9234, s13207, s15850) have been tried with the same parameters as those of normal run except for  $\lambda_2$ .  $\lambda_2$  is assumed to be 0.5 to speed up the simulation (shorter **warmup** period) since these three circuits have a large number of latches. The run time (CPU) is shown in Table 6.

## 6 Conclusions

In this paper we have proposed a statistical estimation technique considering probabilistic delay models for both combinational and sequential CMOS logic circuits. Experimental results show the great impact that the probabilistic behavior of gate delays **can** have on activity of individual nodes as well as power dissipation of a whole circuit. The CPU run time of estimating activity is reasonable and comparable to that of estimating activity without considering uncertainty. Though for our experiments we chose transport delay with inertial delay models, the technique is not restricted to a particular model. When more accurate delay

models are provided, say rise/fall delay models rather than transport ones, our technique can easily adopt the new model. Together with worst-case analysis, the proposed technique can be integrated into a statistical design process that takes advantage of both.

## A Appendix

In Section 3 we **assume** that a bimodal distribution function ( $f(x)$ ) is a linear combination of two normal distribution functions  $f_1(x)$  and  $f_2(x)$ . That is,  $f(x) = P(G_1) \cdot f_1(x) + P(G_2) \cdot f_2(x)$ .  $f_i(\cdot)$  represents the population of  $G_i$ . We will show in this section that  $\frac{\epsilon_b}{\epsilon'}$ , the ratio of the resultant relative error to the user-specified relative error, with the same confidence is,

$$\frac{\epsilon_b}{\epsilon'} \leq \frac{\left(\sqrt{\frac{N_1}{N}} r_m r_{sm} t_{\alpha/2} + \sqrt{\frac{N_2}{N}} z_{\alpha/2}\right) \sqrt{N-1}}{z_{\alpha/2} \sqrt{r_m^2 r_{sm}^2 (N_1 - 1) + (N_2 - 1)}}$$

Assume that in each sample we start to take data after simulating the **sequential** circuit for a **warmup** period. If a total number of  $N$  samples are taken when the **stopping** criterion (**Inequality 2**) is met (the simulation converges), we have  $N_i$  ( $\approx N \cdot P(G_i)$ ) samples collected from  $G_i$ .  $N$  is **determined** by the stopping criterion (**Inequality 2**)

$$N \geq \left(\frac{z_{\alpha/2} s}{m \epsilon'}\right)^2 \quad (14)$$

to meet the user-specified relative error  $\epsilon$  and  $(1 - \alpha) \times 100\%$  confidence level. However, the convergence criterion is based on the assumption that the samples form a **normal distribution** rather than **bimodal**. Therefore, we suspect that the relative error with the same confidence level may not be the same as  $\epsilon$ . Therefore, we will calculate the resultant relative error based on a **bimodal population**. For simplicity, we compare the relative errors in terms of  $\frac{|m-\mu|}{m}$ , denoted as ( $\epsilon'$ ), rather than in terms of  $\frac{|m-\mu|}{\mu}$ , denoted as ( $\epsilon$ ).  $m$  and  $\mu$  are sample mean and true mean, respectively. Notice that the denominators are different in these two expressions.

Let  $s_i$  and  $m_i$  be the sample standard deviation and sample mean for  $G_i$ ,  $s$  be the sample standard deviation for the overall circuit, and  $y_i$  and  $z_j$  be samples from  $G_1$  and  $G_2$  respectively. Hence, we have,

$$\begin{aligned} m_1 &= \frac{\sum_{i=1}^{N_1} y_i}{N_1}, \quad m_2 = \frac{\sum_{i=1}^{N_2} z_i}{N_2}, \quad m = \frac{\sum_{i=1}^{N_1} y_i + \sum_{i=1}^{N_2} z_i}{N} = \frac{N_1 \cdot m_1 + N_2 \cdot m_2}{N}, \\ s_1^2 &= \frac{\sum_{i=1}^{N_1} y_i^2 - N_1 \cdot m_1^2}{N_1 - 1}, \quad s_2^2 = \frac{\sum_{i=1}^{N_2} z_i^2 - N_2 \cdot m_2^2}{N_2 - 1}, \quad \text{and} \\ s^2 &= \frac{\sum_{i=1}^{N_1} y_i^2 + \sum_{i=1}^{N_2} z_i^2 - N \cdot m^2}{N - 1} = \frac{s_1^2 \cdot (N_1 - 1) + s_2^2 \cdot (N_2 - 1)}{N - 1}. \end{aligned} \quad (15)$$

Without loss of generality, we will assume  $P(G_1) \leq P(G_2)$ . Let us consider the case when  $N_1 \leq 30$  and  $N_2 \geq 30$ . This implies that if we use  $N_2$  samples to estimate the true mean value ( $\mu_2$ ) for  $G_2$ , the sample mean  $m_2$  estimated from  $G_2$  is approximately of normal distribution. Therefore,  $\epsilon_2$ , the upper bound on relative error with  $K$  samples ( $K$  calculated by Equation 14) can be calculated by Inequality 1,

$$\epsilon_2 = \frac{z_{\alpha/2} s_2}{m_2 \sqrt{N_2}}. \quad (16)$$

However, if we use  $N_1$  samples to estimate  $\mu_1$ ,  $m_1$  is of *t-distribution* with  $(N_1-1)$  degrees of freedom [20] rather than of normal distribution. Consequently, the upper bound on relative error is

$$\epsilon_1 = \frac{t_{\alpha/2} s_1}{m_1 \sqrt{N_1}}. \quad (17)$$

$z_{\alpha/2}$  and  $t_{\alpha/2}$  are: specific values such that the areas under the normal distribution and t-distribution from  $z_{\alpha/2}$  (or  $t_{\alpha/2}$ ) to  $\infty$  is  $\alpha/2$  Therefore, the overall resultant error is

$$\begin{aligned} \epsilon_b &= \frac{|m - \mu|}{m} = \frac{\frac{N_1}{N}(\mu_1 + m_1 \epsilon_1) + \frac{N_2}{N}(\mu_2 + m_2 \epsilon_2) - (P(G_1)\mu_1 + P(G_2)\mu_2)}{m} \\ &\approx \frac{\frac{N_1}{N} m_1 \epsilon_1 + \frac{N_2}{N} m_2 \epsilon_2}{m} = \frac{\frac{N_1}{N} \frac{t_{\alpha/2} s_1}{\sqrt{N_1}} + \frac{N_2}{N} \frac{z_{\alpha/2} s_2}{\sqrt{N_2}}}{m} \end{aligned} \quad (18)$$

The last equality is derived by substituting  $\epsilon_1$  and  $\epsilon_2$  with Equation 17 and 16. From Inequality 14, we solve for  $m$  by rearranging terms and get  $m \leq \frac{\epsilon' \sqrt{N}}{z_{\alpha/2}}$ . Therefore, equation 18 becomes

$$\epsilon_b \leq \frac{\left(\frac{N_1}{N} \frac{t_{\alpha/2} s_1}{\sqrt{N_1}} + \frac{N_2}{N} \frac{z_{\alpha/2} s_2}{\sqrt{N_2}}\right) \epsilon' \sqrt{N}}{z_{\alpha/2} s} = \epsilon' \frac{\sqrt{\frac{N_1}{N}} t_{\alpha/2} s_1 + \sqrt{\frac{N_2}{N}} z_{\alpha/2} s_2}{z_{\alpha/2} s} \quad (19)$$

Define  $r_m$ ,  $r_{sm1}$ , and  $r_{rsm}$  as  $\frac{m_1}{m_2}$ ,  $\frac{s_1}{m_1}$ , and  $\frac{r_{sm1}}{r_{sm2}}$ , respectively. Therefore,

$$s_1 = r_{sm1} m_1 = (r_{rsm} r_{sm2})(r_m m_2) = (r_{rsm} r_m)(r_{sm2} m_2), \quad (20)$$

$$s_2 = r_{sm2} m_2. \quad (21)$$

Recall that in Equation 15  $s^2 = \frac{s_1^2 \cdot (N_1 - 1) + s_2^2 \cdot (N_2 - 1)}{N - 1}$ . If we substitute  $s_1$  and  $s_2$  with Equation 20 and 21 and rearrange terms, we have

$$\frac{m_2 r_{sm2}}{s} = \frac{\sqrt{N - 1}}{\sqrt{r_{rsm}^2 r_m^2 (N_1 - 1) + (N_2 - 1)}} \quad (22)$$

Similarly, after  $s_1$  and  $s_2$  have been replaced in Equation 19 we have

$$\epsilon_b \leq \epsilon' \frac{\sqrt{\frac{N_1}{N}} t_{\alpha/2} r_{rsm} r_m + \sqrt{\frac{N_2}{N}} z_{\alpha/2} \left(\frac{m_2 r_{sm2}}{s}\right)}{z_{\alpha/2}}. \quad (23)$$

Combining Equation 22 and 23, we derive Inequality 11.

## References

- [1] F.N. Najm, "Transition Density, A New Measure of Activity in Digital Circuits," *IEEE Trans. on Computer-Aided Design*, vol. 12, No. 2, Feb. 1993, pp. 310-323.
- [2] T.-L. Chou, K. Roy, and S. Prasad, "Estimation of Circuit Activity Considering Signal Correlations and Simultaneous Switching," *IEEE Intl. Conf. on Computer-Aided-Design, 1994*, 300-303.
- [3] T.-L. Chou and K. Roy, "Statistical Estimation of Sequential Circuit Activity," *IEEE Intl. Conf. on Computer-Aided-Design, 1995*, to appear.

- [4] T.-L. Chou and K. Roy, "Estimation of Sequential Circuit Activity Considering Spatial and Temporal Correlations," *IEEE Intl. Conf. on Computer Design*, 1995, pp. 577-583.
- [5] A.P. Chandrakashan, S. Sheng, and R. Brodersen, "Low Power CMOS Digital Design," *IEEE Trans. on Solid-State Circuits.*, vol. 27, No. 4, April, 1992, pp. 473-483.
- [6] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," *ACM/IEEE Design Automation Conf.*, 1992, pp. 253-259.
- [7] F.N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Trans. on VLSI Systems*, Dec. 1994, pp. 446-455.
- [8] C.-Y. Tsui, M. Pedram, and A. M. Despain, "Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs," *ACM/IEEE Design Automation Conf.*, 1994, pp. 18-23.
- [9] J. Monteiro, S. Devadas, and B. Lin, "A Methodology for Efficient Estimation of Switching Activity in Sequential Logic Circuits," *ACM/IEEE Design Automation Conf.*, 1994, pp. 12-17.
- [10] R. Burch, F. N. Najm, and P. Yang, T. N. Trick, "A Monte Carlo Approach for Power Estimation," *IEEE Trans. on VLSI Systems*, vol. 1, No. 1, March 1993, pp. 63-71.
- [11] M. G. Xakellis and F. N. Najm, "Statistical Estimation of the Switching Activity in Digital Circuits," *ACM/IEEE Design Automation Conf.*, 1994, pp. 728-733.
- [12] E. Seneta, "Non-Negative Matrices and Markov Chains," 2nd Edition, Springer-Verlag.
- [13] C. Tsui, M. Pedram, and A. Despain, "Efficient estimation of dynamic power consumption under a real delay model," *IEEE Intl. Conf. on Computer-Aided-Design*, 1993, pp.224-228.
- [14] R. Marculescu, D. Marculescu, and M. Pedram, "Efficient Power Estimation for Highly Correlated Input Streams," *ACM/IEEE Design Automation Conf.*, 1995, pp. 628-634.
- [15] R. Marculescu, D. Marculescu, and M. Pedram, "Switching Activity Analysis Considering Spatiotemporal Correlations," *IEEE Intl. Conf. on Computer-Aided-Design*, 1994, pp. 294-299.
- [16] F.N. Najm, S. Goel, and I.N. Hajj, "Power Estimation in Sequential Circuits," *ACM/IEEE Design Automation Conf.*, 1995, pp. 635-640.
- [17] F.N. Najm, M.Y. Zhang, "Extreme Delay Sensitivity and the Worst-Case Switching Activity in VLSI Circuits," *ACM/IEEE Design Automation Conf.*, 1995, pp. 623-627.
- [18] S. G. Duvall, "A Practical Methodology for the Statistical Design of Complex Logic Products for Performance." *IEEE Trans. on VLSI Systems*, Mar. 1995, pp. 112-123.
- [19] A. Papoulis, Probability, "Random Variables, and Stochastic Processes", 3rd Edition, New York: McGraw-Hill, 1991.
- [20] I. Miller and J. E. Freund, "Probability and Statistics for Engineers", Prentice-Hall, 1965.
- [21] E. Cinlar, "Introduction to Stochastic Process," Prentice-Hall, 1975.