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

Tan-Li Chou

Kaushik Roy

School of Electrical and Computer Engineering Purdue University, West Lafayette, IN 47907-1285, USA. tanli@ecn.purdue.edu kaushik@ecn.purdue.edu

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 difficult. 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 delays. 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. Experimental results show that the uncertainty of gate delays makes a great impact on activity at individual nodes (more than 100%) and total power dissipation (can be overestimated up to 65 %) as well.

#### I. 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. The deep submicron processes are also 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 delays. 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 [1], [2], [7], [9], [10], [4], [11], [12], [8]. Some of the 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 [8], [6]. The probabilistic techniques proposed to consider glitches while handling spatial and temporal correlations require building BDDs (Binary Decision Diagrams) of various forms [7], [8], 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 [14]. 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 [7], [11], [12], [13], [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 [14]. This technique uses signal uncertainty to capture the worst casesto 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 this paper, we propose a statistical estimation technique considering uncertainty of gate delays. Since this is a simulation based technique, the spatial and temporal correlations are taken into account automatically. Gate delays with inertial delays are applied in the simulation so that the technique considers spurious transitions without overestimating them. In addition, the inaccuracies of delay models due to sources of uncertainty is captured by the proposed probabilistic delay model. That is, the proposed technique does not try to estimate power dissipation of a *particular* chip. Instead, it gives the average over all the chips of the same circuit under consideration.

The paper is organized as follows. Section II 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 III. Section IV 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

This research was supported in part by ARPA (under contract F33615-95-C-1625), by NSF CAREER award (9501869-MIP), and by IBM Corporation.

presented in section V. Conclusions are given in Section VI.

#### II. PRELIMINARIES AND DEFINITIONS

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

# A. 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  $\Re$  (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  $t \ (\lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} \mathbf{x}(t) dt)$  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  $\lim_{T\to\infty} \frac{n_x(T)}{T}$ , 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.

# B. 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  $POWER_{ave} = \frac{1}{2}V_{dd}^2 \sum_i C_i A(i)$ , 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.

# III. 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 "nearclosed 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. 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,  $a_j = n_j/T$ ,  $j = 1 \cdots K$ , by using different seeds for the random number generators. The sample mean is defined as  $\bar{a} = (\sum_{j=1}^{n} a_j)/K$ . For large K,  $\bar{a}$  will approach the expected value of  $\bar{a}$ , which is  $\lim_{T\to\infty} n_T/T$ , and is denoted as a since the signal at each node is mean-ergodic (section II).  $n_T$  is the number of transitions in the time interval  $\left(\frac{-T}{2}, \frac{T}{2}\right)$ . Similarly, for large K the sample standard deviation s will approach the true standard deviation  $\sigma$ . Furthermore, according to the Central Limit Theorem [16]  $\bar{a}$  is a random variable with mean a and has a distribution approaching the normal distribution if K is large (typically > 30). Likewise  $\sigma \approx s/\sqrt{K}$ . It has been shown in [11], [12] that for  $(1-\alpha)$  $\times$  100% confidence the following inequality holds:

$$\frac{|a-\bar{a}|}{\bar{a}} \le \frac{z_{\alpha/2}s}{\bar{a}\sqrt{K}},\tag{1}$$

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

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

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

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  $\bar{a} \ll 1$ , 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 [12]. 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

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

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

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

In sequential circuits, things are different due to the statebit feedback. One of the approaches is to monitor the statebit probabilities to determine the convergence [13]. 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 by assuming that the state of



Fig. 1. Example: Another STG of a sequential logic circuit.

the machine at time k ( $k^{th}$  clock cycle) becomes independent of its initial state at time 0 as  $k \to \infty$ . That is, the probability will be independent of the initial state as  $k \to \infty$ . However, there are some circuits that can not be directly estimated by this approach. Let us consider the State Transition Graph (STG) of Figure 1. Let  $G_1$  and  $G_2$  denote the set of states given by  $\{s1s0, s1s0\}$  and  $\{s1s0, s1s0\}$ , respectively. Assume 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  $s_{1s_0}(S_0)$  or  $\overline{s_1s_0}(S_1)$ . These sets  $G_1$  and  $G_2$  are called sets of near-closed states. Let y be the output node given by  $y = (s_1s_0 + s_1s_0)x_1$ . Considering only the set of states in  $G_1$ ,  $y = (s_1s_0 + s_1s_0)x_1 = x_1$ , while considering only the set of states in  $G_2$ ,  $y = (s_1s_0 + s_1s_0)x_1 = 0$ . Therefore,  $P(y) = P(x_1)$  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,

$$a(y) = P(G_1) \times a(y|G_1) + P(G_2) \times a(y|G_2).$$
(4)

If STG is given,  $P(G_1)$  and  $P(G_2)$  may be computed by assuming that the primary inputs are either temporally uncorrelated [9], [10] (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  $P(G_2)$  [5]. If we start with 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 close to  $P(G_i)$ . Let  $\epsilon_G$  denote the upper bound on the relative error between the probability and  $P(G_i)$ . Assume that the primary inputs are Markov, the warmup period is empirically determined by [5]  $\ln \frac{2^{i+j}}{\epsilon_G P(G_i)} / ln 1.1$ . Here *i* and *j* are the number of state bits and primary inputs. 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



Fig. 2. Monte Carlo based technique flow chart for both sequential and combinational circuits

 $j^{th}$  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

$$\frac{\left(\sum_{j=1}^{N \cdot P(G_1)} a_j(y|G_1) + \sum_{j=1}^{N \cdot P(G_2)} a_j(y|G_2)\right)}{N} = \frac{N \cdot P(G_1)a(y|G_1) + N \cdot P(G_2)a(y|G_2)}{N} = \frac{P(G_1)a(y|G_1) + P(G_2)a(y|G_2)}{N}$$

which is not biased. The modified version of Monte Carlo based technique for sequential circuits is outlined in Figure 2.

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

# IV. 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.

### A. 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 [15]. 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 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



Fig. 4. A Circuit with random delays

lot variations, operating temperature, power supply voltage, etc. In [14], 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 3. 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 4. As a result, glitches do occur and the values of activities at individual nodes change. This is shown in Figure 4. 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 5. 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 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



Fig. 5. Random delay with truncated normal distribution



Fig. 6. Modified Monte Carlo Based Technique Flow Chart

truncated since it can be considered as a delay fault.

### B. 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 PI. Each component of PI is a stochastic process (see Section II). Therefore, a is also a stochastic process and can be expressed as follows,

$$\mathbf{a} = F(\mathbf{PI}). \tag{5}$$

In Section III, we applied Monte Carlo based techniques to estimate the expected value of a,  $E(\mathbf{a})$ . 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 D is a random vector consisting of all the random variables of gate delays, **a** can be represented as follows,

$$\mathbf{a} = F(\mathbf{PI}, D). \tag{6}$$

Therefore, when applying Monte Carlo based techniques to estimating  $F(\mathbf{PI}, D)$ , delays are modeled as random variables and should be generated from time to time along the 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. Figure 6 outlines the procedure of how the modified Monte Carlo based technique works.

## V. Experimental Results

The Monte Carlo based approach with Probabilistic Delay models (MCPD) to estimate activities at the internal and

TABLE I LONG RUN INFORMATION AND RESULTS OF DIFFERENT DELAY MODELS ON ISCAS combinational benchmark circuits

Ckt	CPU	$\Phi_n$	$\Phi_r$	$\Phi_m$	$\Phi_u$
C432	1.2 hr	130	196	194	215
C499	1.2 hr	184	276	270	270
C880	2.3 hr	279	394	389	391
C1355	5.1 hr	393	805	725	887
C1908	23 hr	625	1446	1515	1634
C2670	30 hr	892	1769	1773	1720
C3540	29 hr	1069	2572	2577	2544
C5315	52 hr	1984	4306	4598	4605
C6288	141 hr	2011	31138	51503	57300
C7552	56 hr	2691	6432	6677	6903

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 IV). 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 is also a random variable that equals to 0.5d. In order to asses 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  $(a_{min} = 0.5)$ . In Table I CPU time for long run MCPD (in hours on a SPARC 5 workstation) is provided. The same table also presents several power dissipation values (normalized power dissipation measure) of different delay models.  $\Phi_n$ ,  $\Phi_r$ ,  $\Phi_m$  and  $\Phi_u$  represent the normalized power dissipation measures of no delays, probabilistic delay models, transport delay (not random but equal to nominal value) with inertial delay being half of the transport delay, and unit delay, respectively, assuming 95% confidence, 5% error, and threshold of  $a_{min} = 0.5$ .

Several interesting observations can be made by examining at Table I. Though the results of unit delay model can have higher spurious transitions (6 out of 10 circuits of Table I), 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 circuits  $\Phi_r > \Phi_m$  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  $a_r$  (activity with probabilistic delay) is greater than  $a_m$  (activity with non-random nominal delay). This is shown in Figure 7. 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



Fig. 8. Uncertainty helping balance paths of circuits (C6288)

spurious transitions (Figure 8). The estimated activity at individual node with nominal delay models can be twice as high as that with probabilistic delay models. As we can see from Table I), the power estimation based on nominal delay models can either underestimate (10 % less) or overestimate ( 65 % more) compared to the results based on probabilistic delay models. That is, the nominal delay model based technique can only estimate the power dissipation of a *particular* chip, while the other one gives an *average* over *all* the chips under consideration. Therefore, if designers or automatic synthesis tools use the estimation of only one chip, it may lead to less optimal design for most of the chips.

The accuracy of the normal run results are shown on Table II. The average relative error (% error) is the average percentage of all the relative errors of individual nodes provided that  $a(y) \ge 0.1$ . 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 [12].

Similarly, for sequential circuits we run MCPD for a long time with 99% confidence, 1.5% error, 0.1 threshold  $(a_{min})$ , near-closed set probability  $(P(G_i))$  of 0.1, and the upper bound on the relative error of  $P(G_i)$  ( $\epsilon_G$ ) being 1%. In contrast to long run MCPD, the normal run MCPD takes the following parameters: 95% confidence, 7.5% error, 0.3 threshold  $(a_{min})$ , near-closed set probability  $(P(G_i))$  of 0.5, with  $\epsilon_G$  being 5%. Comparison between results of long run MCPDand normal run MCPD is shown in Table III. In the table, % error has the same meaning as that of Table II. ND $\Phi$ and PD $\Phi$  represent the normalized power dissipation with zero delay and with probabilistic delay, respectively. No-

TABLE II Individual node information on MCPD in comparison with long run MCPD results

Ckt	CPU	% error	Max abs	a	a~%
	(sec)		error		error
C432	89	1.2	0.052	0.96	5.4
C499	78	1.1	0.034	1.23	2.8
C880	111	1.6	0.032	1.07	3.0
C1355	333	1.3	0.029	2.05	1.4
C1908	807	0.84	0.045	1.07	4.2
C2670	938	0.87	0.038	1.47	2.5
C3540	2140	1.05	0.047	4.60	1.0
C5315	3141	0.86	0.087	3.60	2.4
C6288	10700	0.83	0.343	14.37	2.4
C7552	4989	0.82	0.054	2.64	2.1

TABLE III Individual node information on MCPD in comparison with long run MCPD results

Ckt	#	CPU	$ND\Phi$	$PD\Phi$	% error
	samples	(sec)			
s344	146	119.3	83.8	106.0	0.99
s382	93	41.8	48.5	60.2	1.8
s444	75	23.0	53.2	63.8	6.0
s526	50	16.4	62.4	72.2	7.4
s641	88	225.8	155.5	191.3	1.4
s713	95	250.0	170.1	214.5	1.2
s832	99	137.5	186.5	228.1	1.3
s953	162	260.0	140.3	162.2	2.1
s1196	149	515.1	334.6	420.2	1.1
s1238	134	455.4	345.0	434.4	1.3
s1423	126	605.9	290.1	418.2	2.2

tice that we assume that there are only two near-closed sets. Therefore, the activity sample can be taken from a *bimodal* population  $(G_1 \text{ and } G_2)$  rather than a single population  $(G_1 \text{ or } G_2)$ .

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 number of gates (#gates), primary inputs (#PI), flip-flops (#ff), levels (#levels), and run time (CPU) are shown in Table IV.

TABLE IV 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

# VI. 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.

#### References

- 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, pp. 34-37.
- [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] T.-L. Chou and K. Roy, "Accurate Power Estimation of CMOS Sequential Circuits," *IEEE Trans. on VLSI*, Sep. 1996, pp. 369-380.
- [6] 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.
- [7] 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.
- [8] F.N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," IEEE Trans. on VLSI Systems, Dec. 1994, pp. 446-455.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] F.N. Najm, S. Goel, and I.N. Hajj, "Power Estimation in Sequential Circuits," ACM/IEEE Design Automation Conf., 1995, pp. 635-640.
- [14] F.N. Najm, M.Y. Zhang, "Extreme Delay Sensitivity and the Worst-Case Switching Activity in VLSI Circuits," A CM/IEEE Design Automation Conf., 1995, pp. 623-627.
- [15] S. G. Duvall, "A Practical Methodology for the Statistical Design of Complex Logic Products for Performance." *IEEE Trans. on VLSI Sys*tems, Mar. 1995, pp. 112-123.
- [16] A. Papoulis, Probability, "Random Variables, and Stochastic Processes", 3rd Edition, New York: McGraw-Hill, 1991.