A Statistical Approach to the Estimation of Delay-Dependent Switching Activities in CMOS Combinational Circuits

Yong Je Lim, Kyung-Im Son, Heung-Joon Park, and Mani Soma

University of Washington
Department of Electrical Engineering
BOX 352500
Seattle, WA 98195-2500

Abstract **-- This paper describes a new procedure for estimating the delay-dependent switching activities in CMOS combinational circuits. The procedure is based on analytic and statistical techniques to take advantage of their time-efficiency over conventional logic simulators. Combinational circuits driven by synchronized logic signals are considered as application targets and the statistical properties of logic signals and circuit parameters are defined and evaluated. The experimental result on benchmark circuits shows the significant time efficiency of the proposed procedure.**

1 Introduction

In order to incorporate the reliability assurances at the design stage, which we generally call design-for-reliability techniques, relatively fast but still accurate reliability analysis techniques in comparison to the conventional circuit simulators are required [1-3]. It has been shown that, for CMOS logic circuits, the reliability issues such as power dissipation, electromigration, and the dynamic hot-carrier degradation effects are strongly related to the behavior of the devices and circuits during transitions at the output node. For example, if we can measure the transition rates at a node, the average power dissipation at the node can be estimated by a fairly simple and relatively accurate approximation [4],

$$P_{avg} = \frac{1}{2} C_L V_{dd}^{\ 2} f \qquad (1)$$

where C_L is the load capacitance, V_{dd} is the supply voltage, and f is the transition rate at the node.

A new technique that measures the *transition density* at the circuit nodes was proposed for the approach that the reliability properties can be estimated statistically by measuring the transition rates at the internal and output nodes in combinational circuits [5]. The technique is based on the fact that the transitions at the nodes in combinational circuits are propagated from the primary inputs through viable paths. Two distinctive statistical properties $P(x)$ and $D(x)$ which stand for the *signal probability* and the *transition density*, respectively, are defined and the transition densities at the internal and output nodes in the circuit are *calculated* from those two properties. The major advantage of this technique is that we can calculate the transition densities close to the values that otherwise converge only after lengthy and exhaustive simulations. Later work identified a few drawbacks of the technique such as 1) signal correlations caused by reconvergent fanouts [6], 2) simultaneous switching at the primary inputs [7], and 3) the over-estimation caused by allowing extremely small intertransition time [8].

This research was supported by grants from *LG Electronics Research Center*, Seoul, Korea.

The limitations of previous studies are due to the fact that they mainly concern on generalized combinational circuits. However, by focusing on a specific class of combinational circuits, we are able to model the transitional effects more accurately considering the effects neglected in previous studies. One of the common techniques for high-performance circuit design is to pipeline the significant functional blocks in combinational circuits. In this case, the minimum intertransition time at the primary inputs to the functional blocks is limited by the system clock period. Also, according to the previous studies, node transitions are realized only by transitions originated and propagated from primary inputs. Therefore, the delay-dependent switching activities that we call *glitches* cannot be clearly identified by previous studies.

The purpose of this paper is to propose a new time-efficient statistical technique to estimate the delay-dependent switching activities. By selecting CMOS combinational circuits whose primary inputs are synchronized to a system clock, which is common for pipelining techniques, the proposed technique is more practically meaningful than the analysis on generalized combinational circuits. Also, the extensive use of analytic and statistical method for the estimation can lead to a significant speed up of estimation procedures. Formerly unpredictable switching activities caused by gate and interconnection delays are now tractable without exhaustive simulation runs.

The rest of the paper is organized as follows. In section 2, we identify the characteristics of our model of the combinational circuits, and propose a new statistical approach to estimate delay-dependent switching activities. Experimental results are presented in section 3 where we sought the application of the statistical procedure suggested in section 2 to application circuits. Section 4 gives a brief conclusion of our study.

2 Statistical Modeling of Switching Activities

A. The Circuit

The performance of a combinational circuit is usually measured by the delay between the availability of successive outputs. Therefore, one important requirement of the combinational circuit is the long path timing constraint, which requires that the actual delay of the circuit is not longer than a timing requirement T, usually the clock period [9]. The actual propagation delay of a

DAC 96 - 06/96 Las Vegas, NV, USA

combinational circuit will never be longer than the delay of the longest path in the circuit.

In this study, it is assumed that the circuit is built of logic gates and latches and has the architecture of a synchronous sequential circuit which is well-known as pipelining. Fig. 1 shows a simplified block diagram of the combinational circuit block under consideration. The circuit consists of edge-triggered *latches*

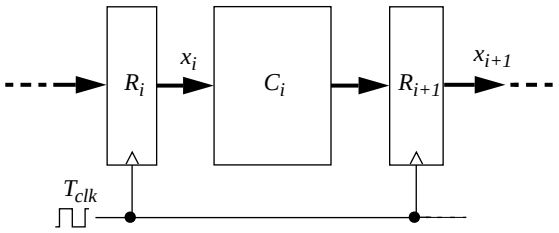

Figure 1. Combinational circuits under consideration.

driven by a common clock and *combinational logic blocks* embedded in the latches.

One of the important advantages of the embedded combinational block model is that the properties of the logic signals sampled at each latches are designed so that they are not to be affected by the propagation delay of the combinational circuit block. The original design concept of this synchronization was to eliminate the faulty effects caused by the propagation delay of the combinational circuit. That is, we can confine the transitional effects such as glitches within each combinational blocks and also within one clock period. In other words, we can decouple the glitches caused by the complex interaction of propagation delay and the circuit topology from the transitions propagated from primary inputs. Because the propagation delays are decoupled from the circuit properties, we can independently analyze the transitions propagated from primary inputs and delay-dependent transitions.

B. Analysis of Delay-Independent Transitions

In this study, we modify the ideal logic signal defined in [5] to incorporate some of the signal characteristics of embedded combinational circuits. First, for the calculation of transitions, we define the primary input signal $x[n]$ which is the output of the latch as:

$$x[n] = x(nT) = x(t)\big|_{t = nT} \tag{2}$$

where $n = (0, \infty]$ is an integer and T is the period of the system clock. The definitions of $P(x)$ and $D(x)$ are now

$$P(x) = \lim_{k \to \infty} \frac{1}{k} \sum_{n=1}^{k} x[n] \text{ , and} \tag{3}$$

$$D(x) = \lim_{k \to \infty} \frac{1}{kT} \sum_{n=1}^{k} |x[n] - x[n-1]| \Big|_{x[0] = x_0} . \tag{4}$$

The relationship between $P(x)$ and $D(x)$ where x is the synchronized logic signal is (Fig. 2):

$$D(x) \le 2 \cdot \tilde{P}(x) \cdot \frac{1}{T} \tag{5}$$

$$\text{where, } \tilde{P}(x) = \begin{cases} P(x), & P(x) \le 0.5 \\ 1 - P(x), & P(x) > 0.5 \end{cases} . \tag{6}$$

Due to the nature of pipelining, the signals sampled at each latches are independent of the logic delays. Therefore, transition densities $D(x_i)$ at the i-th pipeline latch can be calculated by the function:

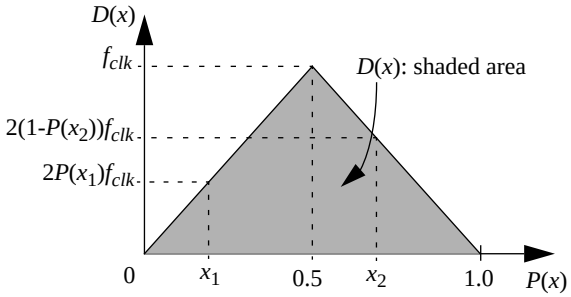

Figure 2. The graphical relations between $P(x)$ and $D(x)$.

$$D(x_i) = \sum_{k=1}^{m} P\left(\frac{\partial x_i}{\partial x_{i-1,k}}\right) \cdot D(x_{i-1,k}) \tag{7}$$

where m is the number of primary inputs at (i-1)th stage. Also, as we decouple the propagation delays from the logic gates, there is no more delay effects at the internal nodes of the circuit block.

C. Analysis of Delay-Dependent Transitions

If we assume no path and interconnection delays, transitions at internal nodes are realized only by the transitions propagated from the primary inputs. The transition rates at internal nodes are weighted according to the functions of the nodes along the path. For the circuit model defined above, whenever the clock triggers the latches, some of the primary inputs will make transitions. Even though the primary inputs of the combinational logic block are updated by the latches, the internal gates in the block may make several transitions before settling to their steady state values for that clock period due to the different path delays. These additional transitions have been called *glitches*. The configuration of the paths is dynamic and the paths are computationally expensive to extract as the size of the sample circuit increases.

In this paper, we suggest a technique to statistically approximate the glitches. If propagation paths have different path delays and the number of the sample paths increases, we can assume that the distribution of path delays can be approximated statistically. There are three properties that cause the glitches at the output of a gate x: a set of input patterns Ω_x, particularly the transition patterns, which is decided by the functionality of the gate, the intertransition time, ζ, between successive input transitions, and the gate delay τ_x. According to our assumptions, the property ζ can be considered as a random variable $\tilde{\zeta}$. These properties must satisfy the following constraints and Fig. 3 shows how the properties, Ω_x, $\tilde{\zeta}$, τ_x, are correlated.

Constraints

(a) Ω_x which cause the glitches at the output node of a logic gate must appear at the input nodes.

(b) $\tilde{\zeta}$ must be larger than the propagation delay of the logic gate, τ_x.

• The constraint (a) only concerns with the patterns at the gate inputs and, therefore, it is a *delay-independent* property. As we assumed in the definition of the circuits, the transitions are synchronized to the system clock and all signals arriving at the input nodes are independent to each other. Considering the constraints and the assumptions, we can estimate the glitch generation rate at the circuit node of x, $D_{glit}(x)$ by defining the following properties.

***Definition 1*:** The *pattern probability* $P_{patt}(x)$ that the glitch-generating patterns appear at the inputs of the gate x at the rising edge of the system clock is

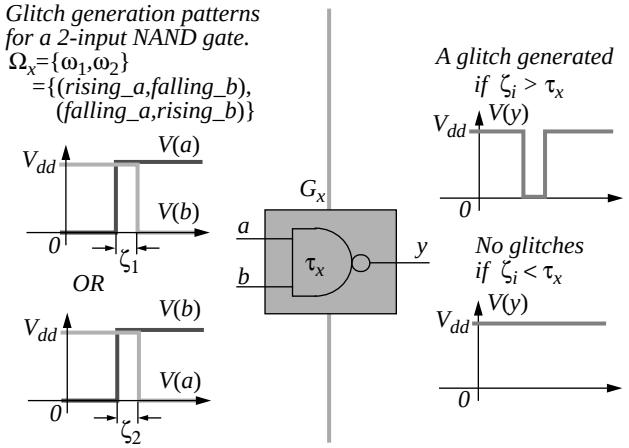

Figure 3. Glitch generation and the related properties for a 2-input NAND gate.

$$P_{patt}(x) = \sum_{i}^{m} P(\omega_i) \cdot P(\omega_{i,\,bias}) \tag{8}$$

where m is the total number of glitch generating patterns, $P(\omega_i)$ is the probability that the i-th pattern $\omega_i \in \Omega_x$ appears at the gate inputs at the rising edge of the system clock, and $P(\omega_{i,bias})$ is the probability that non-transient inputs are biased such that the glitch can go through the logic gate.

In case of the 2-input NAND gate in Fig. 3, for example, we can derive the $P_{patt}(y)$ from the definition 1 as:

$$P_{patt}(y) = P(\omega_1) \cdot P(\omega_{1,\,bias}) + P(\omega_2) \cdot P(\omega_{2,\,bias}) \; . \tag{9}$$

For 2-input NAND gates, the values for $P(\omega_{i,bias})$ always one because there must be at least two input nodes involved to generate a glitch at the output node. The assumption that transitions are synchronized to the system clock gives a hint of the values for $P(\omega_i)$. This leads to the pattern probability as:

$$P_{patt}(y) = \left(\frac{D(a)/2}{f_{clk}} \cdot \frac{D(b)/2}{f_{clk}}\right) + \left(\frac{D(b)/2}{f_{clk}} \cdot \frac{D(a)/2}{f_{clk}}\right)$$

$$= \frac{1}{2} \cdot \frac{D(a) \cdot D(b)}{f_{clk}^{\,2}} \tag{10}$$

where f_{clk} is the frequency of the system clock.

Therefore, the glitch generation rates at the output node x, $\overline{D}_{glit}(x)$, can be represented by:

$$\overline{D}_{glit}(x) = f_{clk} \cdot P_{patt}(x) \; . \tag{11}$$

From the equation (5), for example, the glitch generation rate for the 2-input NAND gate has the upper boundary such that:

$$\overline{D}_{glit}(y) = \frac{1}{2} \cdot \frac{D(a) \cdot D(b)}{f_{clk}} \quad \text{, and}$$

$$\overline{D}_{glit}(y) \le \left(\frac{1}{2} \cdot \frac{1}{f_{clk}} \cdot 2\tilde{P}(a) f_{clk} \cdot 2\tilde{P}(b) f_{clk}\right) \; ,$$

$$\overline{D}_{glit}(y) \le \left(2 \cdot \tilde{P}(a) \cdot \tilde{P}(b) \cdot f_{clk}\right) \; . \tag{12}$$

• From constraint (b), we can formulate the probability that the patterns Ω_x actually generate glitches at the output node. If the intertransition time, ζ_i, between the transitions of the input pattern ω_i, is greater than the propagation delay of the gate x, τ_x, a glitch can overcome the delay-inertia of the gate and can be propagated to the output node of the gate. Again, the path delays from the primary inputs to the internal gate inputs are considered as a random variable $\boldsymbol{d}$ which is assumed to be independent and has the distribution $f(d)$. The boundaries of the distribution, d_{min} and d_{max}, are decided by the path delays of the *shortest path* and the *critical path*, respectively. Both paths are assumed to be *viable*. Considering these assumptions, we define a propagation probability $P_{prop}(x)$.

Definition 2: the *propagation probability* of a gate x, $P_{prop}(x)$, is the probability that a function of the random variable $\boldsymbol{d}$, $\psi(\boldsymbol{d})$, satisfies the following relationship.

$$P_{prop}(x) = P\{|\psi| > \tau\} \; . \tag{13}$$

For the 2-input NAND gate, the function of the random variable, $\psi(\boldsymbol{d})$, is the difference between two path delays d_a and d_b. For convenience, we assign two random variables α and β for $\boldsymbol{d}_a$ and $\boldsymbol{d}_b$, respectively. Then, we can rewrite the equation (13) with the function $\psi(\alpha,\beta)$ as follows.

$$P_{prop}(y) = P\{|\alpha - \beta| > \tau\} \tag{14}$$

$$= 1 - P\{|\alpha - \beta| \le \tau\}$$

$$= 1 - F(\tau)$$

$$= 1 - \iint_{AB} f(\alpha, \beta)\, d\alpha d\beta$$

$$= 1 - \iint_{AB} f(\alpha)\, f(\beta)\, d\alpha d\beta \tag{15}$$

where τ is the gate propagation delay. As we can see in equation (14), path delays and the gate propagation delay have the linear relationship and we can visualize $P_{prop}(y)$ as in Fig. 4.

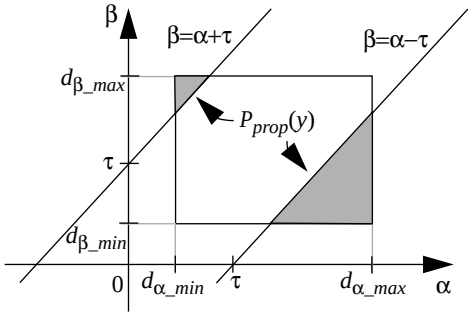

Figure 4. Shaded area indicates the probability, $P_{prop}(y)$.

The challenge in estimating the propagation probability is the proper assignment of appropriate distribution models for the transitions, $f(d)$. Because of the irregularity of Boolean functions in combinational circuits, the distribution model could be a rough measure rather than an accurate prediction. However, as far as the statistical approximations are concerned, we expect to get quite accurate distribution models. To assign the distribution models, we assume that the following two properties are related to the actual distribution. They are 1) *weighting coefficients* on delay paths and 2) *the distribution of delay paths* from primary inputs to a gate.

Weighting coefficients: As we have seen in equation (7), the incoming transitions at an input node x_i of a logic gate are weighted by the coefficient W_i which is:

$$D(y) = W_i \cdot D(x_i) = P\left(\frac{\partial y}{\partial x_i}\right) \cdot D(x_i) \; , \tag{16}$$

where y is the output node of the gate. W_i is normally represented by a product of $P(x_j)$'s or $P(x_j)$'s where x_j is one of the inputs to the gate except x_i. Therefore, W_i can be approximated in the form of

$$W_i = \prod_{(j=1),\, j \neq i}^{m} P_j \approx K^{m-1} \quad (17)$$

where m is the number of inputs to the gate, P_j is the signal probability at a input node x_j, which is either $P(x_j)$ or $P(\bar{x}_j)$, and K is the average signal probability. The output y will be one of the inputs to the next stage and the transition $D(y)$ will be successively weighted by the weighting coefficients of each gates as shown in Fig. 5. Therefore, we can approximate the weighting coefficient

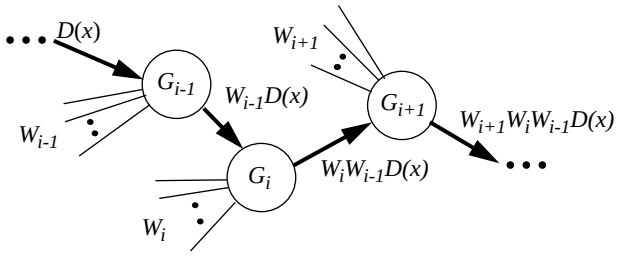

Figure 5. Weighting coefficients on a delay path

on a single delay path d_i as:

$$W_{d_i} = \left(K^a\right)^{s(d_i)} \quad (18)$$

where a is the average number of inputs except the one on the delay path d_i and $s(d_i)$ is the depth on the delay path.

Distribution model: The distribution of delay paths is assumed to be *Gaussian*. According to the central limit theorem, as the size of samples becomes large, the distribution of samples converges to Gaussian. All transitions are propagated through the delay paths and each path is weighted by the coefficient *W*. By combining weighting coefficients and the Gaussian distribution together, we define the *parametric distribution model* of the transitions as:

$$f(d) = G \cdot K^{a \cdot s(d)} \cdot N(m, \sigma) \; . \quad (19)$$

In the above, G is the normalization factor, K is the normalized signal probability associated with the gate of interest, a is the average fan-in of the gate under consideration, $s(d)$ is the average depth when the path delay is d, and $N(m, \sigma)$ is the *Gaussian* distribution with mean m and standard deviation σ [13]. The major advantage of this parametric model is that we can estimate the distribution with simple parameters. Also, calculations rather than exhaustive simulation runs are all we need to extract the parameters. However, one of the possible drawbacks is the calculation time for parameters m and σ, because, if the application circuit has lots of reconvergent fan-outs, the extraction could be *NP-complete*. However, as we have assumed in the beginning, if the circuit has few number of reconvergent fan-outs, we still can manage to calculate the parameters in reasonable time span.

Physically, $P_{prop}(x)$ is a weighting coefficient that shows *how the gate propagation delay affects the glitch generation rates*. By weighting the glitch generation rate $\bar{D}_{glit}(x)$ with $P_{prop}(x)$, we can finally estimate the glitch generation rate $D_{glit}(x)$ which is,

$$D_{glit}(x) = P_{prop}(x) \cdot \bar{D}_{glit}(x) \; . \quad (20)$$

Again, for the 2-input NAND gate, by combining equation (11) and (20), we estimate the glitch generation rate at the output node of the gate as:

$$max(D_{glit}(y)) = \frac{1}{2} \cdot P_{prop}(y) \cdot \frac{D(a) \cdot D(b)}{f_{clk}} \; . \quad (21)$$

Based on two definitions and the equation (20), we develop a procedure to estimate the transition rates including glitch generation rates at the internal nodes of the combinational circuits. The procedure consists of two functional modules, one is to calculate the transition rates and pattern probabilities, while the other module is to calculate the propagation probability at the circuit nodes. Analysis results of each functional modules are combined together to estimate the transition rates. Fig. 6 shows the overall flow of the proposed procedure.

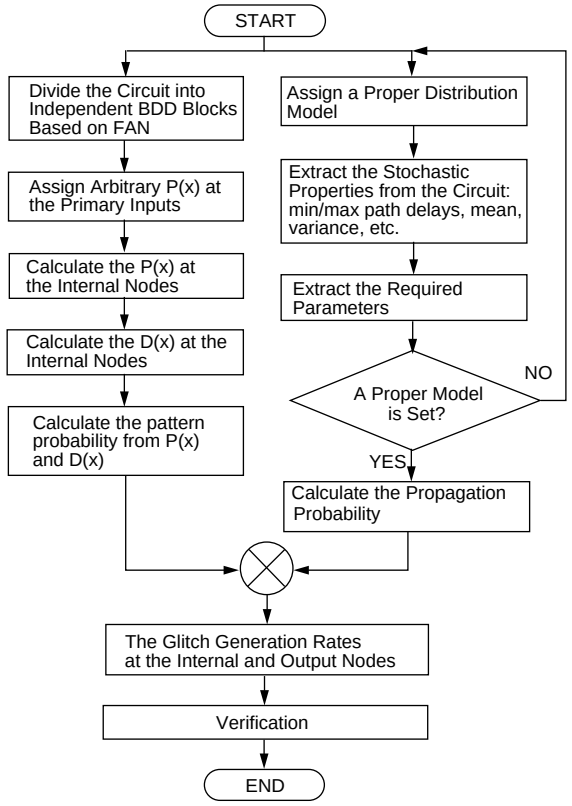

Figure 6. A block diagram of the glitch estimation procedure.

3. Experimental Results

For a simple verification of the suggested procedure in Fig. 6, we complete the estimation of the transition rates at the output node of the 2-input NAND gate. The distribution models for random variables α and β, $f(\alpha)$ and $f(\beta)$, are assumed to be uniform distributions with minimum and maximum boundaries as:

$$f(\alpha) = \frac{1}{\alpha_{max} - \alpha_{min}} \cdot \{U(t - \alpha_{min}) - U(t - \alpha_{max})\} \quad , \quad (22)$$

$$f(\beta) = \frac{1}{\beta_{max} - \beta_{min}} \cdot \{U(t - \beta_{min}) - U(t - \beta_{max})\} \quad . \quad (23)$$

Table 1 shows the parameter values assigned for the *Verilog*-HDL simulation and calculation of the probabilities. For the logic

Table 1: Parameters assigned for the case study

Signal	d_{min} (*nsec*)	d_{max} (*nsec*)	P
a	3.0	13.0	0.5
b	5.0	10.0	0.5

simulation, input patterns are generated by the uniform random number generation function in *Matlab* and the path delays for both inputs are assigned by using the built-in probabilistic distribution functions of the *Verilog*-HDL. The simulation has run for 300,000 clocks with various gate delays, τ, ranging from 0.1 to 8 *nsec*. *Matlab* programming utilities are also used to calculate $P_{patt}(y)$

and $P_{prop}(y)$. Fig. 7 shows the result of both simulation and calculation. The normalized error between two results is less than

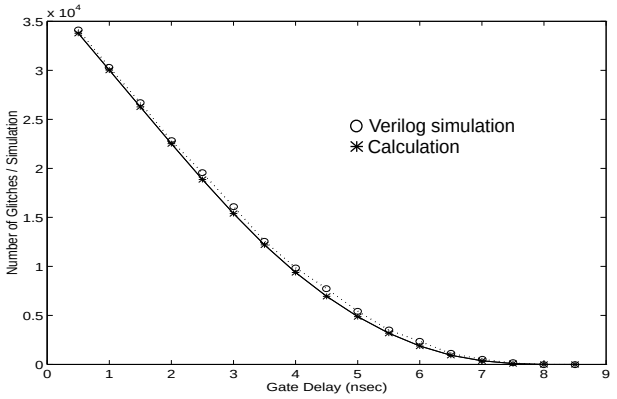

Figure 7. Glitch rates vs. propagation delay of the logic gate

0.13% and the error is mostly due to the precision of the *Verilog* simulation. As we have seen in Fig. 4, no glitches must be generated if the gate delay exceeds 8 *nsec*, that is because the propagation probability is zero beyond that point, and both results well satisfy this condition. If we consider the gate propagation delay of the 2-input NAND gate which is normally less than 1 *nsec* [10], the number of glitches which is more than 13.5% of the output transition is a significant number considering the power dissipation and the hot-carrier reliability problem.

For more complex applications of the procedure, ISCAS-85 benchmark circuits are considered. One of them is c3540 which is a combinational network functioning as an ALU and control circuitries. The circuit contains 1669 logic gates and 50 primary inputs and 22 primary outputs. To calculate the pattern probabilities, we need to extract $P(x)$ and $D(x)$. For this purpose, we convert the circuit to Boolean modules. The Boolean modules are especially useful for BDD applications for extracting the $P(x)$ and $D(x)$. Unfortunately, the BDD's are unacceptably large for the whole circuit, and, therefore, we partition the circuit into 355 smaller Boolean modules based on FAN [11]. Fig. 8 is the sample $P(x)$'s at the internal nodes of the circuit. The difference between

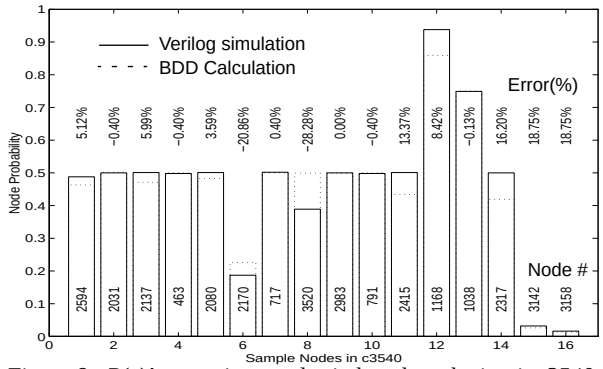

Figure 8. $P(x)$'s at various nodes in benchmark circuit c3540.

the results of the calculation and simulation run is mostly due to reconvergent fanouts which cause correlation between signals. However, that is the trade off between accuracy and speed. The pattern probabilities for the gates of interest are calculated based on the signal probabilities.

To verify our assumptions on the parametric distribution model, we first calculate the required parameters for the model. For this purpose, ISCAS-85 benchmark circuits are transformed to tree-structured descriptions and all physical paths and their propagation delays are extracted. The viability of the paths is not considered and the gate delays are assigned based on the CMOS portable library by VLSI Technology, Inc.[10] The average fan-in for the nodes of interest and the average number of stages vs. the path delays for each circuits are calculated from the description. Also, to calculate the mean and standard deviation of the path delays at the nodes of interest, we extract all possible paths and their delays. Verification of the assumption, that the distribution of path delays is Gaussian, is performed by reconstructing Gaussian distribution using the mean and standard deviation that we calculated and comparing the reconstructed distribution to the distribution of the extracted path delays. The overlapped distributions in Fig. 9 show fairly good matches between two distributions.

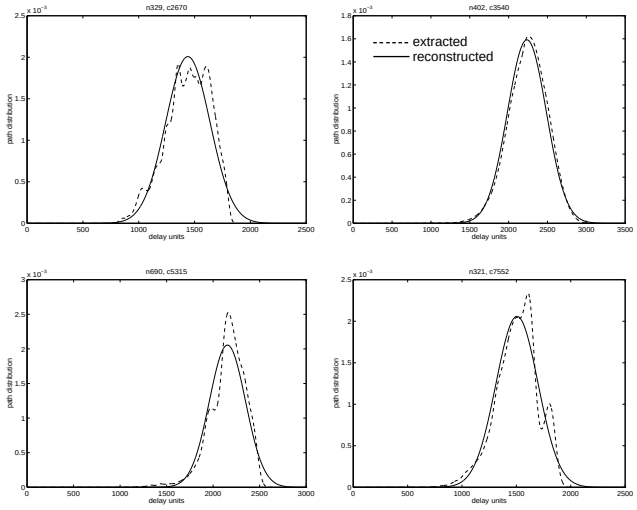

Figure 9. The distribution of the delay paths for the benchmark circuits.

The validation of the calculated parameters is performed by comparing the distribution of transitions to the parametric distribution model. We first run the HDL simulations on ISCAS-85 benchmark circuits and the transitions at certain nodes are recorded. Again, the gate delays are assigned using the same CMOS library used for the path extraction. By combining the recorded transitions with the derating factors such as the changes of the operating temperature and supply voltage, and the process variation, we could reveal the pseudo-real distribution of the transition at the circuit nodes. Again, we reconstructed our parametric distribution models based on the calculated parameters. Two distributions are compared in Fig. 10 and we found that the range of the distributions are all matched between the two. As the characteristics of the circuit, such as delay paths and reduction rates, have more statistical tendencies, there is less distortion of the distribution curves. One of the extreme cases where the statistical assumptions failed is the distribution for c1908 in Fig. 10. Although the ranges of the delay are closely matched for both distributions, the distribution of actual transitions severely deviates from it's statistical counter part. The cause of the deviation in this case is the lack of delay paths at one of the input nodes associated with the gate output. As long as the statistical assumptions hold, reconstructed parametric distributions approximate the real distributions with little deviation.

Finally, we select the circuit nodes which best represent the statistical characteristics such as large number of delay paths and nodes along the paths. With the pattern probabilities and the parametric distributions for the circuit nodes in selected benchmark circuits, especially c3540, c5315, and c7552, we calculate the glitch generation rates and compare them with the

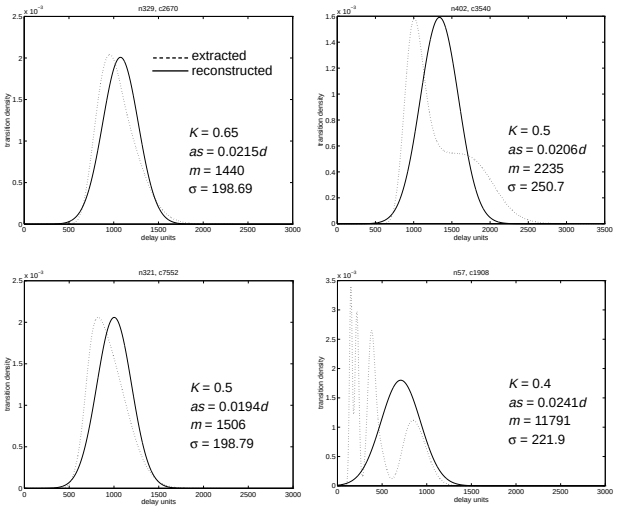

Figure 10. The distribution of the transition.

results of simulation runs. Table 2 shows the results on selected circuit nodes. Benchmark circuits with small number of gates are

Table 2: Estimated glitches on various circuit nodes after 50,000 clock cycles. (sampled from c3540, c5315, & c7552)

node	gate type	P_{prop}	trans.	glitch (sim.)	glitch (est.)	err (%)	glit./ tran (%)
n402	NAND	0.9092	30529	5679	5197	8.4	18.6
n403	NAND	0.9103	20464	2857	2724	4.7	13.9
n404	NAND	0.9103	21134	2918	2724	6.6	13.8
n2725	AND	0.8056	17769	2433	2346	3.6	13.7
n3657	OR	0.9815	28134	3031	3422	12.2	10.8
n1520	NAND	0.8732	18825	3051	2750	9.9	16.2
n1521	NAND	0.8758	30676	5735	4961	13.5	18.7
n2855	NAND	0.9256	18613	2707	2900	7.1	14.5
n2856	NAND	0.8883	18855	3132	2784	11.1	16.6
n4092	NAND	0.9073	18743	2837	2864	0.95	15.1
n320	NAND	0.8830	30265	5321	4965	6.7	17.6

not considered for the final verification because the nodes in those circuits cannot be assumed to have proper statistical characteristics. Three types of switching activities are listed in the table: 1) '*transitions*' propagated from primary inputs, 2) '*glitches*' counted on *simulation* runs, and 3) '*glitches*' estimated by the proposed *estimation* procedure. As we can see in the column of the absolute difference between simulation and estimation results, the glitches estimated by the proposed procedure are closely matched to the values extracted after long simulations. Though we used *Matlab* utilities for the most of the calculations which is a lot slower than sophisticated programs, we still managed to get the result much faster than *Verilog*-HDL simulation runs. The circuit nodes with higher difference between simulation results and the estimated values are mostly caused by the signal correlation between gate inputs.

One important observation from the result is that unexpected transitions which are responsible for more than 20% (one glitch contributes two transitions) of total switching activities at the circuit nodes in the form of glitches are generated. Though our assumption is that the generated glitches are not propagated through to the successive delay paths, some of them are actually propagated through the paths until they are filtered out by one of logic gates along the paths. Those propagated glitches add up to generated glitches and they could be responsible for more unexpected switching activities at the circuit nodes. Without a fast analysis method to identify the delay-dependent switching rates, we could end up with underestimated power assumption and poor reliability predictions.

Our approach is limited by the assumptions made in the statistical calculations. We have assumed fanout-free circuits and the independence between primary inputs. But, what causes the dependency between paths are reconvergent fanouts in circuitry. Even though the primary inputs are independent, there is no guarantee that the signals at the internal nodes are independent.

4 Conclusion

We have proposed a new time-efficient procedure to estimate the delay-dependent switching activities in CMOS combinational circuits. By focusing on a specific class of combinational circuits, we were able to model the transitional effects more precisely. Also, extensive use of analytic and statistical methods for the estimation showed significant time-efficiency over conventional logic simulation. The results of the application on experimental circuits showed that both transitions propagated from primary inputs and unexpected glitches generated by the interaction between gate functions and interconnection delays were well-tractable by proposed statistical technique.

References

[1] P. Yang and J. Chern, "Design for reliability: the major challenge for VLSI," *Proc. of the IEEE*, vol. 81, no. 5, pp. 730-744, May, 1993.

[2] R. H. Tu, E. Rosenbaum, C. C. Li, W. Y. Chan, P. M. Lee, B-. K. Liew, J. D. Burnett, P. K. Ko, and C. Hu, *BERT-Berkeley Reliability Tools*, Memorandum no. UCB/ERL M91/107, Berkeley, 1991.

[3] C. Hu, S. C. Tam, F. C. Hsu, P. K. Ko, T. Y. Chan, and K. W. Terrill, "Hot-electron induced MOSFET degradation model-model, monitor, and improvement," *IEEE Trans. on Electron Devices*, vol. ED-32, no. 2, pp.375-385, Feb. 1985.

[4] H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 468-473, Aug. 1984.

[5] F. N. Najm, "Transition density: a new measure of activity in digital circuits," *IEEE Trans. on CAD*, vol. 12, no. 2, pp.310-323, Feb. 1993.

[6] M. G. Xakellis and F. N. Najm, "Statistical estimation of the switching activity in digital circuits," *Proc. of 31st ACM/IEEE Design Automation Conference*, pp. 728-733, 1994.

[7] T. Chou, K. Roy, and S. Prasad, "Estimation of circuit activity considering signal correlations and simultaneous switching," *Proc. ICCAD*, pp. 300-303, 1994.

[8] F. N. Najm, "Improved estimation of the switching activity for reliability prediction in VLSI circuits," *Proc. CICC*, pp. 17.7.1-17.7.4, 1994.

[9] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, New York, Computer Science Press, 1990.

[10] *VSC470 CMOS Portable Library*, VLSI technology, Inc., April, 1992.

[11] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithm," *IEEE Trans. on Computers*, vol. 27, no. 3, pp. 265-272, March, 1983.