

Determining Accuracy Bounds for Simulation-Based Switching Activity Estimation

Anthony M. Hill and Sung-Mo (Steve) Kang
University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering
Coordinated Science Laboratory
1308 W. Main Street
Urbana, IL 61801

Abstract

Discrete simulation based methods for switching density estimation can provide very accurate results, but a number of unresolved problems such as estimating the required number of input patterns impedes the widespread usefulness of this technique. In this paper, we address the issue of obtaining run-time and a-priori estimates of the number of input patterns required for a specified accuracy. We obtain these estimates through the definition of a set of multinomial random variables and a set of functions based on the parameters of these random variables. Experimental results indicate that the proposed techniques provide accurate estimates of the number of patterns required for a specified switching density accuracy.

I. Introduction

Accurate estimation of switching density for both power dissipation and reliability analysis is a significant concern for future integrated circuits. With battery-powered applications destined for continued future growth, the issue of extending the useable lifetime is becoming increasingly significant. Recent advances in battery technology may help diminish this problem, but they are not keeping pace with the growth trend in integrated circuit size. Thus, in many state-of-the-art portable applications, functionality is sacrificed for reduced power consumption typically in the form of a power supply reduction. Besides this problem, the continual increase in packing density facilitates an increasing number of devices per chip which can seriously impact reliability. The mean time-to-failure may be significantly lower if proper techniques are not employed to handle high switching activity regions.

Recently, a number of novel low-power techniques have been proposed. However, before many of these techniques can be used, accurate estimates of switching density must be obtained. The identification of high switching density regions ("hot spots") is crucial to improving the reliability since these areas tend to fail prematurely due to higher average stress directly correlated to switching density.

Furthermore, since power consumption is directly proportional to transition density, regions with high switching activity consume a significant portion of the total power budget, making them prime candidates for power reducing optimizations.

Two methods are principally used for calculation of switching density: probabilistic analysis and simulation-based analysis [6]. In probabilistic analysis signal statistics such as transition density are propagated from primary inputs to primary outputs [5]. This technique has the benefit of low computational complexity but suffers from reduced accuracy. Extensions of probabilistic analysis have been proposed which sacrifice speed for accuracy. However, the run time required for these methods begins to approach that of discrete simulation methods (for example, see the run times of [3] and [7]). Simulation-based analysis estimates switching density by applying a large number of patterns to the circuit and directly observing the switching rate using a discrete logic simulator. While this can provide very accurate results, a number of unresolved issues inhibit the use of this technique.

One of the principal problems with simulation-based estimation is determining the number of patterns to be applied to the logic circuit. To the circuit designer, it is desirable to estimate the number of patterns and hence, the run time required for simulation. This allows for adjustment of the specified accuracy to suit both simulation error bounds and time limitations. Qualitatively, the larger the number of input patterns applied the better the resulting accuracy, but thus far, few techniques have been proposed which can approximate the relationship between accuracy and the number of applied patterns.

Two methods to obtain confidence intervals in switching density and power estimation have been proposed in the literature based directly on the central limit theorem [3,4]. These methods can provide stopping criteria for a specified accuracy but require run-time information. Stopping criteria for the power consumption of an entire circuit is proposed in [4] where it is assumed that the sampled power consumption follows a normal distribution. Stopping criteria to obtain a specified switching density

accuracy at all individual nodes is proposed in [3]. This method is based on the fact that the distribution of the mean approaches a normal distribution for a large number of samples [9]. This technique requires observing N samples each of duration T since the standard deviation of an unknown distribution must be estimated. This can require significantly more samples being taken than necessary. Furthermore, neither of the above methods can be used to obtain information on the required number of patterns before simulation.

As a precursor to this work, a method for estimating the number of required patterns in an a-priori fashion has been proposed in [8]. That methodology relies on the assumption that each input pattern generates at most one rising event on each node. Although this assumption is valid for a high percentage of nodes in a circuit, it can not guarantee that the specified accuracy will hold for all nodes. Experimentally, we find that the results from [8] can provide accurate stopping criteria for ~80% of the nodes in our largest benchmark circuit.

In this paper, we present a technique that allows a-priori *estimation* of the number of patterns required to meet a given error tolerance with a specified confidence. We also show how this technique may be used early in simulation to provide a more accurate estimation of the total run time and may be used to provide a stopping criterion. This technique is applied to estimate the number of patterns required for gate-level switching density accuracy. Gate-level estimation is crucial for diagnosing high-power logic nodes for power optimization and for determining potential reliability problems such as electromigration and hot-carrier degradation caused by high switching activity. We say that a circuit is gate-level accurate when the measured switching density at each node is within a specified error with a specified confidence. To achieve a gate-level error of at most 0.05 with a confidence of 99% (represented herein as (0.05,0.99)). Then for every gate we desire an absolute error of <0.05 with 99% confidence, that is 99 out of 100 estimates for that gate should satisfy the error bound.

The remainder of this paper is organized as follows. In section II, we present an approximation of the signal statistics. This enables us to derive a relationship between the number of patterns applied and the resulting accuracy in section III. Section IV contains experimental results which justify our approach, and in section V, we present our conclusions.

II. Approximation of Signal Statistics

In this section we introduce a method of representing the effect of an applied pattern in terms of Bernoulli random variables (RVs). Upon application of a set of n patterns,

we will form two multinomial RVs. This allows us to use statistical techniques to estimate confidence intervals on the parameters of these RVs in section III.

Suppose that a single input vector is applied to a logic circuit. For each node, k , in the logic circuit we define two sets of mutually exclusive and exhaustive events:

$$F_i^{(k)}(R_i^{(k)}) = \begin{cases} 1 & \text{if } i \text{ falling (rising) transitions occur} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $i=0,\dots,t$ and t represents the maximum number of transitions observed. As an example, $F_2^{(k)}=1$ for node k if and only if exactly two falling transitions are observed on that node. For simplicity, the superscript (k) will henceforth be omitted, but it should be kept in mind that each node has associated with it a set of unique events. It can be easily seen that the events $F_i(R_i)$ and $F_j(R_j)$ are mutually exclusive since the observation of i falling (rising) transitions on a specific node precludes the simultaneous observation of j falling (rising) events on that node for all $i \neq j$. This set of events is also exhaustive since application of a single input pattern results in the observation of at most t falling (rising) transitions. One may wonder how t scales with circuit size. For all benchmark circuits tested, the maximum t observed has been <10. In general, we anticipate that t will scale logarithmically with circuit size although circuits can be constructed where t scales linearly with the number of gates.

With each event $F_i(R_i)$, we can associate a random variable, $\mathbf{F}_i(\mathbf{R}_i)$. We define the probability that exactly i falling (rising) events are observed upon the application of a single input pattern as $p_{Fi}=\mathcal{P}\{\mathbf{F}_i=1\}$ ($p_{Ri}=\mathcal{P}\{\mathbf{R}_i=1\}$) where \mathcal{P} denotes probability. In other words, when a single pattern is applied, we expect that with probability p_{Fi} (p_{Ri}) exactly i falling (rising) transitions will be observed on the node associated with the event $F_i(R_i)$. Independently sampling the above RVs by applying n input patterns to the circuit yields two multinomial RVs for each node, one for rising transitions and the other for falling transitions. We denote these RVs as $\mathbf{M}_F(\mathbf{M}_R)$ with multinomial probability density functions (pdfs)

$$f(x_0, \dots, x_t) = \frac{n!}{x_0! x_1! \dots x_t!} p_0^{x_0} p_1^{x_1} \dots p_t^{x_t}, \quad (2)$$

where p_i is the value p_{Fi} (p_{Ri}) and x_i is the number of times exactly i falling (rising) transitions occur on the node of interest during the n random samples. Of course this pdf requires that $\sum x_i = n$ and $\sum p_i = 1$ are satisfied.

In switching activity estimation, we apply n input patterns and look at the distribution of the number of rising and falling transitions on each node. The total number of rising transitions on a node may differ from the number of falling transitions by at most one since each rising transi-

tion requires a corresponding falling transition except possibly on the final transition. Thus, if i falling transitions are observed for an applied input pattern then the number of rising transitions observed on that node is either $i-1$, i , or $i+1$. Let us assume that each of these three possibilities is equiprobable. Then for a large number of samples $p_{Fi} \approx p_{Ri}$ which allows us to approximate $\mathbf{M}_F \approx \mathbf{M}_R$ for each node. In other words, the number of times we observe exactly i rising transitions will be approximately the same as the number of times we observe exactly i falling transitions. Therefore, instead of considering separate rising and falling distributions, we will only consider a single distribution \mathbf{M} with corresponding switching probabilities p_i . Without loss of generality, in the remainder of this paper we will think of \mathbf{M} in terms of observed rising transitions.

III. Bounding Transition Density Error

In the previous section, we showed how a multinomially distributed RV, \mathbf{M} , may be defined for each node in the logic circuit. From the definition of this random variable, we can calculate the switching density and the number of patterns that should be applied for a specified accuracy. The switching density, s , on a node is given in transitions per clock cycle and is estimated by

$$\hat{s} = 2 \cdot \sum_{i=1}^t i \cdot \frac{x_i}{n}, \quad (3)$$

where x_i is the observed number of times a single input pattern generates exactly i rising events on the node, n is the total number of applied patterns, and the multiplicative factor of 2 accounts for rising and falling transitions. The term x_i/n is an estimate for the parameter p_i which we denote \hat{p}_i .

Our goal is to approximate the required value of n such that the estimated switching density is within ϵ of the actual value with confidence $1-\delta$ for any particular node in the circuit, i.e.,

$$\mathcal{P}\{|\hat{s} - s| \leq \epsilon\} \geq 1 - \delta. \quad (4)$$

For the development of the method we present, we have assumed that an absolute error accuracy is required. However, if we desire a specified accuracy in percentage error, $(\epsilon', 1-\delta)$, we can use (4) and the modified value of ϵ ,

$$\epsilon = \frac{\epsilon'}{1 - \epsilon'} \hat{s} \quad (5)$$

This imposes no additional computational burden since \hat{s} is already computed during simulation. Note that a given percentage error in switching density accuracy corresponds to the same percentage error in power accuracy.

We begin our derivation by computing the variance of (3). Then we employ the central limit theorem to obtain the

desired relationship between accuracy and the number of applied patterns. From the relation [1],

$$\sigma^2 = E[s^2] - E[s]^2 \quad (6)$$

we can write the variance of \hat{s} as

$$\sigma^2 = \frac{4}{n} \left\{ \sum_{i=1}^t i^2 \cdot p_i - \left(\sum_{i=1}^t i \cdot p_i \right)^2 \right\}. \quad (7)$$

This can be consistently estimated by substituting \hat{p}_i for p_i since the true p_i are not known. For a sufficiently large value of n , the central limit theorem may be used to obtain [1,2],

$$\frac{\hat{s} - s}{\sigma} \xrightarrow{d} N(0,1), \quad (8)$$

i.e., $(\hat{s} - s)/\sigma$ approaches a standard normal distribution in density. If we define $z_{1-\delta/2}$ as the $100 \times (1-\delta/2)$ th percentile of the standard normal distribution, we obtain

$$\mathcal{P}\{|\hat{s} - s| \leq \sigma \cdot z_{1-\delta/2}\} \approx 1 - \delta. \quad (9)$$

This can easily be manipulated to yield the minimum number of required patterns as a function of the parameters ϵ and δ as

$$n \approx \frac{4 \cdot z_{1-\delta/2}^2}{\epsilon^2} \left[\sum_{i=1}^t i^2 \cdot \hat{p}_i - \left(\sum_{i=1}^t i \cdot \hat{p}_i \right)^2 \right]. \quad (10)$$

This relationship between accuracy and the number of required patterns can be used to assess the relative run times of alternative values for $(\epsilon, 1-\delta)$. If we know or measure the required number of patterns, n_1 , for a specified accuracy, $(\epsilon_1, 1-\delta_1)$, then for a second accuracy, $(\epsilon_2, 1-\delta_2)$, the number of patterns required is approximated by

$$n_2 \approx \frac{\epsilon_1^2}{\epsilon_2^2} \cdot \frac{z_{1-\delta_2/2}^2}{z_{1-\delta_1/2}^2} \cdot n_1. \quad (11)$$

In deriving (11), it has been assumed that the maximum number of rising transitions, t , and the values for \hat{p}_i remain invariant for different accuracy parameters. Usually a value of $n_1=100$ is sufficient to obtain a reasonably accurate estimate for these quantities. This result yields one potential accuracy-driven simulation algorithm. First, simulation is performed with loose error bounds, say $(0.2, 0.8)$, with the stopping criteria determined by (10) being satisfied at all nodes. Then, the entire accuracy vs. run time tradeoff curve can be determined and appropriate bounds can be selected with respect to run times and/or error goals.

Given run-time information, (10) provides a stopping criterion for a specified accuracy. However, there is considerable interest in estimating the number of required patterns in an a-priori fashion without requiring some type of initial simulation. The only unknown values in (10) are

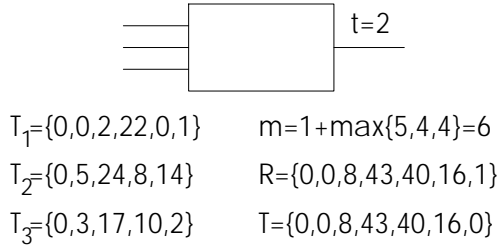


Fig. 1. Computation of T for a three-input gate.

the measured values for \hat{p}_i and the maximum number of transitions observed, t . If we can find realistic estimates for the values of t and \hat{p}_i , these can be used to estimate the number of patterns required. We now present an algorithm that has been empirically found to produce good estimates for these parameters.

First, we need to obtain the maximum number of transitions, t . For this, we assume that all gates have equal unit delays. Let $T^{(k)}$ be a vector with entry j representing the number of paths from the primary inputs to the output of gate k which could generate a transition at time j . Furthermore, let $m^{(k)}$ represent the latest time at which gate k 's output will switch and $t^{(k)}$ be the maximum number of rising transitions on gate k .

The first step in this portion of our algorithm is to initialize the primary inputs. We assume that all inputs transition at time zero, i.e., $T_0^{(p)}=1$. This method does not depend on this assumption; however, for the average input switching density to be >0.5 , transitions at times other than zero are required under the unit-delay model. Next, the gates are topologically ordered, and the above values are propagated as demonstrated in Fig. 1 according to the following equations:

$$\begin{aligned}
 m^{(k)} &= 1 + \max_i \{m^{(i)}\} \\
 R_j^{(k)} &= \begin{cases} \sum_i T_{j-1}^{(i)} & \text{if } j > 0 \\ 0 & \text{otherwise} \end{cases} \\
 T_j^{(k)} &= \begin{cases} R_j^{(k)} & \text{if } j > 0 \text{ and } R_j^{(k)} / \sum_n R_n^{(k)} > \lambda \\ 0 & \text{otherwise} \end{cases}, \quad (12) \\
 t^{(k)} &= \left\lceil \frac{1}{2} \cdot \sum_j \left\{ 1 \text{ if } T_j^{(k)} > 0 \right\} \right\rceil
 \end{aligned}$$

where $i=1,\dots,r$, $j=n=0,\dots,m^{(k)}$, and r is the number of inputs to gate k . After these values are computed for all gates, t is estimated as the maximum value of $t^{(k)}$. If the percentage of transitions in the temporary variable, $R_j^{(k)}$, is not at least λ (empirically $\lambda=1\%$), then we assume that no events will actually be observed on the output at time j . We have found that including this condition on the propagation of

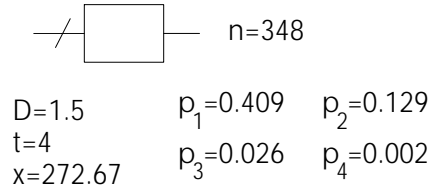


Fig. 2. Calculation of n .

events from input to output dramatically improves the estimate of $t^{(k)}$. Physically, a transition on an input signal may not propagate to the output due to the functionality of the gate and signal correlation. However, to achieve run-time efficiency, we can not directly consider these effects but instead account for them by the λ parameter.

Once we have an estimate for t , we need to determine the values for \hat{p}_i . To achieve this, we first call a probabilistic simulator based on [5]. This provides us with a quick estimate for the switching density, D , at each node. We then use $D^2/4$ to approximate the second inner-term in (10). In addition, we can use D to estimate the probabilities according to the equations:

$$\begin{aligned}
 \hat{p}_i &= \frac{(t+1-i)^4}{x + \sum_{j=1}^t (t+1-j)^4} \\
 D &= 2 \cdot \sum_{k=1}^t k \cdot \hat{p}_k
 \end{aligned} \quad (13 \text{ a,b})$$

where x corresponds to the weighting assigned to the probability that zero transitions occur. The value of x is computed by substituting (13a) into (13b). This value is then resubstituted in (13a) to calculate the values for \hat{p}_i . Finally, these quantities are used together with (10) to produce the final result. This yields,

$$n = \frac{Dz^2}{7\epsilon^2} \cdot \frac{8t^4 + (32-14D)t^3 + (56-42D)t^2 + (48-21D)t + 7D-4}{2t^3 + 6t^2 + 3t-1} \quad (14)$$

The use of (13) and (14) is demonstrated in Fig. 2 where n is calculated for (0.1,0.9). Notice that this model may produce a negative value for x if D is sufficiently large compared to t . However, this has not been observed in any benchmark circuits. If it occurs, then a cubic, quadratic, linear, or constant dependence can be tried in (13) until a non-negative value for x is obtained.

IV. Experimental Verification

In this section we present data verifying our approach to estimating the value for n in both a-priori and run-time paradigms. We begin by presenting data certifying the validity of our variance estimate in (10). Next, we assess the accuracy of (12) and (14) for a-priori estimation. Finally, we demonstrate the use of (11) in producing an accuracy vs. run-time tradeoff curve.

Table 1. Number of patterns for various $(\epsilon, 1-\delta)$.

| Circuit | Gates | Number of Patterns Required | | |
|---------|-------|-----------------------------|-------------|--------------|
| | | (0.1,0.9) | (0.05,0.99) | (0.01,0.999) |
| b9 | 168 | 336 | 3302 | 134463 |
| C432 | 364 | 625 | 6138 | 250118 |
| C880 | 538 | 593 | 5819 | 237311 |
| alu2 | 610 | 659 | 6465 | 265325 |
| C1355 | 703 | 663 | 6511 | 263724 |
| C499 | 719 | 525 | 5157 | 210099 |
| C1908 | 962 | 633 | 6218 | 253319 |
| alu4 | 1137 | 713 | 7002 | 285334 |
| C2670 | 1180 | 1147 | 11257 | 459016 |
| C3540 | 1656 | 1591 | 15613 | 636699 |
| C5315 | 2794 | 1452 | 14247 | 581073 |
| C7552 | 3877 | 2058 | 20195 | 823587 |

To validate our approach we have selected a benchmark suite that consists of circuits from ISCAS 1985 and the 1989 MCNC International Workshop on Logic Synthesis. These circuits are simulated via a gate-level simulator with delay models based upon capacitive loading and signal slew rates. For all simulations we specify that each primary input signal has a probability of 0.25 of changing logic value at the beginning of each clock cycle and that the input slope is randomly distributed over $[0ns, 2ns]$. Finally, the clock period is set to be the maximum delay of the circuit plus 1ns. For verification purposes we need to first obtain “exact” values for the switching density of each node. To acquire these values an initial simulation with an accuracy specification of (0.01,0.999) was performed. Table 1 contains a list of the benchmark circuits and the number of patterns required to satisfy this accuracy specification as well as two other representative accuracy specifications.

To verify the accuracy of (10) we simulated each benchmark circuit 1000 times with an accuracy of (0.1,0.9) and then calculated the maximum violation rate over all nodes in the circuit. A violation occurs at a node if $|\bar{s} - s| > \epsilon$, and the violation rate is defined as the number of violations divided by the total number of simulations. In this instance, we expect the maximum violation rate to be approximately $\delta=0.1$, and as indicated by Table 2, this is indeed the case with the maximum observed violation rate over all benchmark circuits being 15%. We should not expect the actual violation rate to be below 10% in all circuits for a number of reasons. In deriving the approximate confidence intervals used in (10), we utilized the central limit theorem. While this theorem provides us with a limiting distribution for $n \rightarrow \infty$, our simulations only required 300-2000 patterns which may introduce a small amount of error. In addition, 1000 simulations may not be adequate to obtain a true measure of the violation rate. Therefore, a few percent error in the observed $1-\delta$ for this approximation is quite acceptable, and in any case, we may compensate for it through slightly tighter accuracy specifications.

Table 2. Violation rate for (0.1,0.9).

| Circuit | Maximum Violation Rate | % Nodes in Violation |
|---------|------------------------|----------------------|
| b9 | 8.6% | 0.00% |
| C432 | 11.5% | 1.10% |
| C880 | 11.5% | 0.74% |
| alu2 | 10.0% | 0.00% |
| C1355 | 12.0% | 0.57% |
| C499 | 14.8% | 1.39% |
| C1908 | 13.0% | 0.73% |
| alu4 | 10.0% | 0.00% |
| C2670 | 15.0% | 0.59% |
| C3540 | 9.0% | 0.00% |
| C5315 | 9.5% | 0.00% |
| C7552 | 7.5% | 0.00% |

Next, we consider the error of our a-priori pattern number estimation algorithm given principally by (12) and (14). For each benchmark circuit, data was collected during the calculation of “exact” results on the maximum number of transitions observed for each node. Next, 100 simulations were performed with an accuracy specification of (0.1,0.9) and the mean number of patterns required was recorded. Finally, (12) and (14) were used to predict these values for each circuit with the results being presented in Table 3. In this table, the ratio of the predicted n to the observed n is tabulated to facilitate comparison across the accuracy spectrum since the ratio will remain constant for all accuracies. As this data indicates, our estimate for t is in excellent agreement with the observed value for t in all benchmark circuits. Furthermore, the number of patterns predicted is within an acceptable range of the measured number for all benchmarks as borne out by the ratio of the two. The only circuit for which our algorithm significantly underestimates the number of required patterns is C3540. This indicates that for larger circuits our quartic relationship between probabilities may need to be revised if slight underestimation is to be avoided. In general, as circuit size increases the data indicates that our combined approximations become better and produce more accurate results.

In many instances it is desirable to obtain the entire accuracy vs. run time tradeoff curve using (11). To demonstrate this technique, we consider application to C1908. Initially simulation was performed with an accuracy specification of (0.2, 0.8). This required a total of $n_1=428$ patterns where the stopping criterion was (10) being satisfied for all 962 nodes. We can use this result and (11) to plot the accuracy vs. number of patterns tradeoff curve as shown in Fig. 3. As an insert to this figure the tradeoff between ϵ and $1-\delta$ is depicted for $n=600$ patterns. This insert clearly indicates that it requires relatively fewer patterns to increase $1-\delta$ than to increase ϵ . This figure also clearly shows the inverse quadratic dependence on ϵ evident in (10).

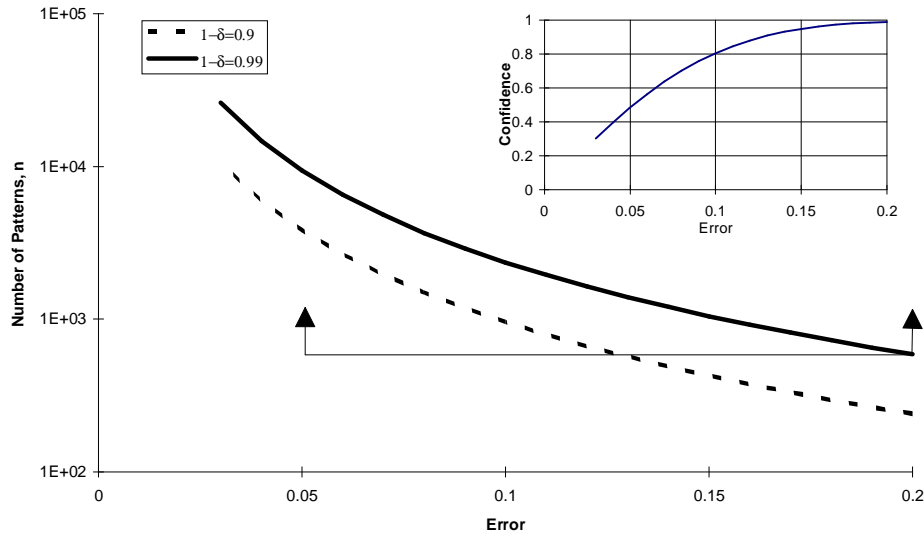


Fig. 3. Number of patterns required as a function of accuracy for C1908.

V. Conclusions

For reliability assessment and power analysis, estimation of the switching density for individual nodes is essential. Switching density estimation by discrete simulation can yield very accurate results but is hampered by an ambiguous relationship between the number of patterns applied and the accuracy of the resulting estimates.

In this paper, we have presented an analysis of the relationship between the number of applied patterns and the resulting accuracy of the estimated densities. We have provided an algorithm that allows for a-priori estimation of the number of required patterns for a specified accuracy. In addition, we have shown how our methodology may be used in conjunction with run-time data for error control during simulation. Furthermore, we have shown how one can obtain the entire accuracy vs. run time tradeoff curve using this methodology in a very efficient manner. In the future, our techniques may be extended to analyze the re-

lationship between the number of applied patterns and total circuit power.

References

- [1] J. Blum, J. Rosenblatt, *Probability and Statistics*, Philadelphia, W. B. Saunders Company, 1972.
- [2] P. Meyer, *Introductory Probability and Statistical Applications*, 2nd ed., Reading MA, Addison-Wesley, 1970.
- [3] M. Xakellis, F. Najm, "Statistical estimation of the switching activity in digital circuits," *31st. Design Automation Conference*, pp. 728 - 733, 1994.
- [4] R. Burch, F. Najm, P. Yang, T. Trick, "A monte carlo approach for power estimation," *IEEE Trans. on VLSI Systems*, vol. 1, no. 1, pp. 63-71, March 1993.
- [5] F. Najm, "Transition density: A new measure of activity in digital circuits," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 12, no. 2, pp. 310-323, February 1993.
- [6] F. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Trans. on VLSI Systems*, vol. 2, no. 4, pp. 446-455, December 1994.
- [7] R. Marculescu, D. Marculescu, M. Pedram, "Switching activity analysis considering spatiotemporal correlations," *Proceedings ICCAD '94*, pp. 300-303, 1994.
- [8] A. M. Hill, S.-M. Kang, "Accuracy bounds in switching activity estimation," *Custom Integrated Circuits Conference*, 1994.
- [9] I. Miller, J. Freund, *Probability and Statistics for Engineers*, 3rd ed., Englewood Cliffs NJ, Prentice Hall, 1985.

Table 3. Accuracy of the estimation of t and n for $\epsilon=0.1$, $1-\delta=90\%$.

| Circuit | Measured | | Calculated | | Ratio |
|---------|----------|------|------------|------|-------|
| | t | n | t | n | |
| b9 | 3 | 336 | 4 | 467 | 1.39 |
| C432 | 4 | 625 | 7 | 1525 | 2.44 |
| C880 | 5 | 593 | 7 | 1740 | 2.93 |
| alu2 | 5 | 659 | 10 | 1690 | 2.56 |
| C1355 | 5 | 663 | 5 | 644 | 0.97 |
| C499 | 4 | 525 | 4 | 668 | 1.27 |
| C1908 | 5 | 633 | 7 | 1098 | 1.73 |
| alu4 | 6 | 713 | 10 | 1758 | 2.47 |
| C2670 | 7 | 1147 | 8 | 1828 | 1.59 |
| C3540 | 7 | 1591 | 7 | 1197 | 0.75 |
| C5315 | 7 | 1452 | 9 | 2320 | 1.60 |
| C7552 | 7 | 2058 | 11 | 2900 | 1.41 |