Statistical Estimation of the Switching Activity in Digital Circuits†

Michael G. Xakellis and Farid N. Najm

Coordinated Science Lab. & ECE Dept.
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Abstract–Higher levels of integration have led to a generation of integrated circuits for which power dissipation and reliability are major design concerns. In CMOS circuits, both of these problems are directly related to the extent of circuit switching activity. The average number of transitions per second at a circuit node is a measure of switching activity that has been called the *transition density*. This paper presents a statistical simulation technique to estimate individual node transition densities. The strength of this approach is that the desired accuracy and confidence can be specified up-front by the user. Another key feature is the classification of nodes into two categories: regular- and low-density nodes. Regular-density nodes are certified with user-specified *percentage error* and confidence levels. Low-density nodes are certified with an *absolute error*, with the same confidence. This speeds convergence while sacrificing percentage accuracy only on nodes which contribute little to power dissipation and have few reliability problems.

I. Introduction

The advent of VLSI technology has brought new challenges to the manufacture of integrated circuits. Higher levels of integration and shrinking line widths have led to a generation of devices which are more sensitive to power dissipation and reliability problems than typical devices of a few years ago. In these circuits excessive power dissipation may cause run-time errors and device destruction due to overheating, while reliability issues may shorten device lifespan. It is especially useful to diagnose and correct these problems before circuits are fabricated. In CMOS circuits, gates draw current and consume power only when making logical transitions. As a result, power dissipation and reliability strongly depend on the extent of circuit *switching activity*. Hence, there is a need for CAD tools that can estimate circuit switching activity during the design phase.

† This work was supported in part by the National Science Foundation (NSF), under grant MIP-9308426.

Circuit activity is strongly dependent on the inputs being applied to the circuit. For one input set the circuit may experience no transitions, while for another it may switch frequently. During the first input set the circuit dissipates little power and experiences little wear, but for the second its activity could cause device failure. However, the specific input pattern sets cannot be predicted up-front. Furthermore, it is impractical to simulate a circuit for all possible inputs. Thus, this input pattern dependence severely complicates the estimation of circuit activity.

Recently, some approaches have been proposed to get around this problem by using probabilities to represent *typical* behavior at the circuit inputs. In [1], the average number of transitions per second at a circuit node is proposed as a measure of switching activity, called the *transition density*. An algorithm was also proposed to propagate specified input transition densities into the circuit to compute the densities at all the nodes. The algorithm is very efficient, but it neglects the correlation between signals at internal nodes. This leads to errors in the individual node densities that may not always be acceptable, especially since the desired accuracy cannot be specified up-front.

This correlation problem was avoided in [2], where the total average power of the circuit (a weighted sum of the node transition densities) was statistically estimated by simulating the circuit for randomly generated input patterns. The power value is updated iteratively until it converges to the true power with a user-specified accuracy (*percentage error* tolerance), and a user-specified confidence level. It was found that convergence is very fast because the distribution of the overall circuit power was very nearly Gaussian and very narrow about its mean.

While power estimation is one important reason to find the transition densities in a circuit, it is not the only one. The densities can be used to estimate average current in the power and ground busses, to be used for electromigration analysis. For this application, it is not enough that the overall estimated power be accurate, but the individual node density values must be accurate as well. However, it becomes extremely inefficient to apply the statistical sampling technique in [2] to single gates (so as to estimate the transition density at every gate output). This is because a large number of input patterns is required to converge for nodes that switch very infrequently, as we will demonstrate later on.

In this paper, we will present an extension of the approach in [2] whereby we remove the above limitation and efficiently estimate the transition density at all circuit nodes. To overcome the slow convergence problem, we apply *absolute* error bounds to nodes with low transition density values, instead of percentage error bounds. This is done by establishing a threshold, $\eta_{\min}$, to classify node transition density values. Any node with a transition density value less than the threshold is classified as a *low-density* node and is certified with absolute error. Nodes with transition density values equal to or above the threshold are classified as *regular* density nodes and are certified with a percentage error. A major advantage of this approach is that the desired accuracy can be specified up-front by the user. Furthermore, the percentage error bound is relaxed (i.e., replaced by an absolute error bound) *only* on low-density nodes. These nodes dissipate little power and have few reliability problems.

The statistical simulation techniques to be presented are implemented in a prototype called "Mean Estimator of Density" (MED). MED's performance is evaluated by looking at the accuracy of its results, its convergence rate, and its execution time.

The paper is organized as follows. In the next section, the statistical estimation technique is described. Section III presents experimental data and evaluates MED's performance, while section IV presents a summary.

II. Proposed solution

This section presents our statistical estimation technique for computing the transition densities at all circuit nodes. It is expected that the user will supply the transition density, denoted $D(x)$, for every circuit input node. Actually, the user should also specify the fraction of time that a circuit input signal is high, called the probability at that node, and denoted by $P(x)$. If unspecified, these probabilities can be assigned default values of 1/2. This technique, as well as the other techniques reviewed in the introduction, apply only to combinational circuits. It can be applied to sequential circuits provided that the transition densities at the latch outputs are specified.

Given the input transition densities and probabilities, we can use a random number generator to generate corresponding logic input waveforms with which to drive a simulator. Based on such a simulation of the circuit for a given time period T, we can count the number of transitions at every node, a number which will be called a *sample* taken at that node. If we repeat this process N times, and form the average $\bar{n}$ of the number of transitions at a node, so-called the *sample mean*, then $\bar{n}/T$ is an estimate of the transition density at that node.

It is well known from statistical *mean estimation* [3] that for large values of N, the sample mean $\bar{n}$ will approach the true average number of transitions in T, to be represented by η. Likewise, the sample standard deviation s will approach the true standard deviation σ for large N. One continues to take samples (make simulation runs) until $\bar{n}$ is *close enough* to η. The method by which one tests for this is called the stopping criterion, to be discussed next. The following sub-section details the mechanism of input waveform generation.

A. Stopping Criterion

According to the *Central Limit Theorem* [3], $\bar{n}$ is a value of a random variable with mean η whose distribution approaches the *normal distribution* for large N. The minimum number of samples, N, to satisfy near-normality is typically 30. It is also known that for such values of N one may use s as an estimate of σ.

Since the distribution of sample means is near-normal, we can make inferences about the quality of an *individual* sample. With $(1-\alpha)$ *confidence* it then follows that [3]:

$$-z_{\alpha/2}\sigma \leq \eta - \bar{n} \leq z_{\alpha/2}\sigma \tag{1}$$

where $z_{\alpha/2}$ is defined so that the area to its right under the standard normal distribution curve is equal to $\alpha/2$.

Equation (1) may be rearranged to better accommodate mean estimation, by using:

$$\sigma \approx \frac{s}{\sqrt{N}} \tag{2}$$

which is justified for values of N which normalize the sample mean distribution, typically for $N \geq 30$. This is not restrictive; typical simulations take many more samples. The transformed equation is more applicable to our problem, so that with confidence $(1-\alpha)$, we have:

$$\frac{|\eta - \bar{n}|}{\bar{n}} \leq \frac{z_{\alpha/2}s}{\bar{n}\sqrt{N}} \tag{3}$$

If ϵ_1 is a small positive number, and if

$$N \geq \left(\frac{z_{\alpha/2}s}{\bar{n}\epsilon_1}\right)^2 \tag{4}$$

samples are taken, then ϵ_1 places an upper bound on the percentage error of the sample with $(1-\alpha)$ confidence:

$$\frac{|\eta - \bar{n}|}{\bar{n}} \leq \frac{z_{\alpha/2}s}{\bar{n}\sqrt{N}} \leq \epsilon_1 \tag{5}$$

This may also be expressed as the percent deviation from the population mean η:

$$\frac{|\eta - \bar{n}|}{\bar{n}} \leq \epsilon_1, \text{ translates to } \frac{|\bar{n} - \eta|}{\eta} \leq \frac{\epsilon_1}{1-\epsilon_1} = \epsilon \tag{6}$$

where ϵ is defined to be a user-specified error tolerance. Thus (4) provides a stopping criterion to yield the accuracy specified in (6) with confidence $(1-\alpha)$.

It should be clear from (4) that for small values of $\bar{n}$, say $\bar{n} < \eta_{\min}$, the number of samples required can become too large. It thus becomes too expensive to guarantee a percentage accuracy for low-density nodes. Instead, we can certify these nodes with an absolute error bound, as follows. Suppose we use the modified stopping criterion:

$$N \geq \left(\frac{z_{\alpha/2}s}{\eta_{\min}\epsilon_1}\right)^2 \tag{7}$$

for low-density nodes (with $\bar{n} < \eta_{\min}$). Then with $(1-\alpha)$ confidence:

$$|\eta - \bar{n}| \leq \frac{z_{\alpha/2}s}{\sqrt{N}} \leq \eta_{\min}\epsilon_1 \tag{8}$$

Thus $\eta_{\min}\epsilon_1$ becomes an *absolute* error bound that characterizes the accuracy for low-density nodes.

We therefore classify the circuit nodes into regular-density nodes and low-density nodes. During the algorithm (after N exceeds 30) (4) is used as a stopping criterion as long as $\bar{n} \geq \eta_{\min}$, otherwise (7) is used instead. The value of $\eta_{\min}$ can be specified by the user and strongly affects the speed of the algorithm, as will be shown in section III.

Although the percentage error for low-density nodes sharply increases as $\bar{n} \rightarrow 0$, the absolute error remains relatively fixed. In fact, it can be shown that the absolute error bounds for low-density nodes are always *less than* the absolute error bounds for regular density nodes. Although these nodes require the longest time to converge, they have the least effect on circuit power and reliability. Therefore the above strategy reduces the execution time, with little or no penalty.

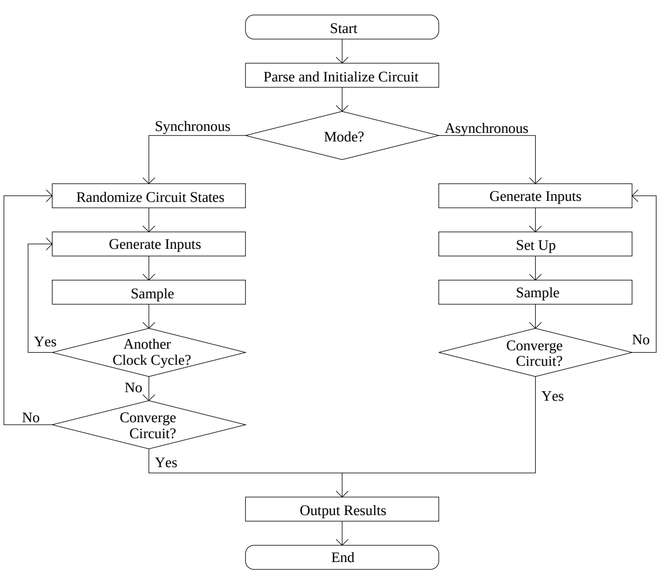

Fig. 1. MED block diagram.

B. Input Generation

Fig. 1 illustrates the simulator block diagram, and shows that it can run in one of two modes, synchronous and asynchronous. In the synchronous mode, we assume that the (combinational) circuit is part of a larger synchronous sequential circuit design, so that its input events should be generated in synchrony. Otherwise, asynchronous operation is assumed and events do not have to be synchronized. Thus the only difference between synchronous and asynchronous operation is the generation of input transitions driving the circuit.

In the synchronous mode, an input node may transition only at the beginning of a clock cycle, so that the input pulse widths are discrete multiples of the clock period, T_c. The *distribution* of the high (and low) pulses at the inputs is arbitrary, and can be user-specified. Our implementation assumes that an input signal is *Markov*, so that its value after a clock edge depends only on its value before the clock edge, once that value is specified, and not on its values during earlier clock cycles. Under this assumption, it can be shown that the pulse widths have a *geometric* distribution. If μ_0 and μ_1 are the mean low and high pulse widths, computed from [1] as:

$$\mu_1 = \frac{2P(x)}{D(x)} \tag{9}$$

$$\mu_0 = \frac{2\left[1 - P(x)\right]}{D(x)} \tag{10}$$

then it can also be shown that the probability that a low signal will transition high on the clock is:

$$P(1 \mid 0) = \frac{T_c}{\mu_0} \tag{11}$$

and the probability of a high signal transitioning low on the clock is:

$$P(0 \mid 1) = \frac{T_c}{\mu_1} \tag{12}$$

A random number generator uses (11) and (12) to generate input transitions for every clock cycle.

For circuits running asynchronously, input transition generation proceeds differently. Since input transitions may occur at any time, the input generation routine determines the length of time between transitions instead of the probability of transition at the clock edge. Again, the distribution of the pulse widths is arbitrary, and can be specified by the user. Our implementation was based on a Markov assumption, so that the length of time between successive transitions is a random variable with an *exponential* [3] distribution. The length of time a signal stays in the low (high) state has mean μ_0 (μ_1). From this information, the waveform is easily generated.

Additionally, when running asynchronously the simulator requires a *setup period*. This is a waiting period during which no samples are collected. It is needed for the same reasons that a setup period was required in [2]. Briefly, it allows the circuit to "get up to speed." Before sampling begins, transitions at the inputs must be allowed to propagate into the internal nodes of the circuit. Until all levels of the circuit are involved, switching activity is artificially low and any power or reliability estimates will be skewed. The length of the setup period should be, as was also shown in [2], no less than the maximum delay of the circuit.

III. Experimental results

This technique has been implemented in the program MED (Mean Estimator of Density), in which the basic simulation capability is event-driven, gate level, with a scalable delay timing model (based on output capacitance and fanout). In general, any simulation strategy can be used, so that the technique presented can be *wrapped around* any existing simulator and simulation library. In this section we present data collected with MED, and show that it is both accurate and practical on a number of large benchmark circuits.

A. Input Specification

The experimental results to be presented are based on a specification of the *typical* circuit inputs as follows.

In the synchronous mode, we assumed that the circuit would be operated near its maximum operating frequency, so that the clock cycle time, T_c, is close to the maximum circuit delay, T_{max}. Unless otherwise specified, the results presented were based on a value of T_c that is 1 nsec longer than T_{max}.

The second assumption concerns the input probability and transition density values. It was specified that every input node has probability of 1/2 and a transition density of $1/(2T_c)$. Thus, on average, each input node was assumed to spend an equal time high and low, and to have one transition every other clock cycle.

Finally the transition density values were normalized to the clock period, i.e., the transition densities output by the program are expressed in terms of transitions per clock cycle. The output densities are then invariant to clock cycle time, and the user has a more intuitive view of circuit activity - 0.5 transitions per clock cycle is much more informative than 5e7 transitions per second. This is especially useful in light of the fact that the absolute transition density varies linearly with clock frequency.

Asynchronous input probability and density assumptions are similar to the synchronous assumptions. Inputs are assumed to have probabilities of 1/2 and transition densities of $1/(2T_{max})$. Transition densities for asynchronous circuits are normalized by T_{max}.

B. Data Collection

The issues to be investigated are (1) the error of the technique, (2) the handling of low-density nodes, and (3) the practicality of the technique for large circuits. The data collected should allow MED's performance to be evaluated in the above three categories.

B.1. Establishing accurate transition density values

The first step in evaluating MED's performance is to establish a set of *accurate* node transition densities. This baseline would then be used to calculate the actual error of the estimated transition density values. This was done by running MED for a long time on the benchmark circuits presented at ISCAS in 1985 [4]. Typically, in order to achieve 99.99% confidence and 1% error tolerance for all the nodes, this required millions of input vectors and hours or days of SUN Sparc-10 CPU time. Table I lists the circuits, number of gates, number of samples, and execution times for each circuit and mode of operation.

TABLE I
Long run information

circuit	#gates	Synchronous Mode		Asynchronous Mode	
		#samples	run time	#samples	run time
c432	160	1677390	1.5 h	606900	57.7 min
c499	202	588870	35.2 min	285200	30.2 min
c880	383	1161840	2.3 h	813700	2.7 h
c1355	546	1051250	3.9 h	335200	1.5 h
c1908	880	2281460	11.7 h	1001200	11.7 h
c2670	1193	1592660	14.2 h	748300	12.0 h
c3540	1669	1556380	16.5 h	1514000	27.9 h
c5315	2307	1373840	24.7 h	831200	28.9 h
c6288	2406	444620	42.3 h	262700	68.4 h
c7552	3512	1390320	59.4 h	1008500	45.6 h

B.2. Calculating error distributions

To verify that MED produces results within the specified error tolerances, 10 runs with $\eta_{\min}$ varying linearly from 0.05 to 0.50 were executed with 95% confidence ($1-\alpha = 0.95$) and 5% error tolerance ($\epsilon = 0.05$) on the ISCAS 1985 set. Node transition density values from the runs were compared with the standard values computed above. Regular transition density values, $\bar{n} > \eta_{\min}$, are valid if 95% of the values have less than 5% error. Low-density values, $\bar{n} < \eta_{\min}$, are valid if 95% of the values satisfy $|\eta - \bar{n}| \leq \eta_{\min}\epsilon_1$.

Tables II and III give the percentage of transition density values out-of-bounds for all the circuits under investigation. From the tables it can be seen that this percentage is very low, well below the specified 5%. This happens because many of the nodes are oversampled, since the simulator will run until the last node converges. This yields more accuracy than what is actually specified by the user.

TABLE II

Performance in synchronous mode

circuit	η_{min}	%regular-density nodes out-of-bounds	%low-density nodes	%low-density nodes out-of-bounds
c432	0.35	1.17	12.69	0.00
c499	0.05	0.00	13.11	0.00
c880	0.20	0.00	13.74	1.64
c1355	0.15	0.21	17.69	0.00
c1908	0.10	0.00	11.27	1.94
c2670	0.45	0.18	16.58	0.00
c3540	0.10	0.00	9.77	0.00
c5315	0.45	0.00	15.49	0.78
c6288	0.40	0.00	13.68	0.90
c7552	0.40	0.03	7.77	1.04

B.3. Comparison of η_{min} and execution time

It is expected that since the simulator runs until its last node converges, and further that low-density nodes require the longest time to converge, then adjusting η_{min} would significantly affect overall simulation time while sacrificing percentage accuracy on a small number of nodes.

Ten simulations are run with η_{min} varying linearly from 0.05 to 0.50. SUN Sparc-ELC execution times in cpu seconds are tabulated and reported in Table IV. Low-density nodes typically require the largest number of samples to converge, and as a result execution time drops dramatically as η_{min} rises. In some cases however, the lowest-density nodes are not the last to converge, and the adjustment of η_{min} has no effect on execution time.

TABLE III

Performance in asynchronous mode

circuit	η_{min}	% regular-density nodes out-of-bounds	% low-density nodes	%low-density nodes out-of-bounds
c432	0.40	0.00	9.14	0.00
c499	0.10	0.49	16.39	2.50
c880	0.10	0.97	6.98	0.00
c1355	0.15	0.41	17.52	0.97
c1908	0.45	0.00	14.00	3.91
c2670	0.45	0.18	16.21	0.91
c3540	0.25	0.07	21.40	0.00
c5315	0.45	0.00	15.33	0.52
c7552	0.45	0.03	6.86	1.18

The simulation times for all circuits except for c6288 follow a general downward trend, as shown in Fig. 2. The curves result from averaging circuit execution times (excluding c6288) normalized by the time required for the circuit to simulate with $\eta_{min} = 0.05$.

The behavior of circuit c6288 is an exception to this trend. The execution times for c6288 are essentially invariant to η_{min} for $0 < \eta_{min} < 0.5$. This occurs because c6288 has regular density nodes with considerable variation, and at least one of the regular density nodes with $\bar{n} > 0.5$ converges after all low-density nodes. Because of this, the last nodes to converge are not affected by η_{min}.

TABLE IV

Execution times in cpu seconds, on a sun sparc elc, with varying η_{min}

circuit	synchronous execution times for η_{min} =										asynchronous execution times for η_{min} =									
	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
c432	89	99	60	60	61	49	44	23	18	18	111	95	110	102	103	105	97	80	70	61
c499	271	90	90	38	25	25	28	14	14	14	367	295	62	67	58	75	69	56	67	45
c880	729	366	228	131	130	101	82	69	53	53	947	685	438	352	293	231	180	167	126	107
c1355	609	200	207	186	128	109	88	85	85	86	1717	718	394	248	249	241	261	307	241	263
c1908	1978	741	789	316	294	285	172	155	152	150	4892	2434	1444	1431	789	800	726	749	741	526
c2670	2911	1276	1222	899	643	564	461	466	387	358	2764	2486	2112	1553	1243	974	923	902	885	799
c3540	4579	2458	2130	1146	883	736	732	729	667	465	6268	3814	3396	3619	2917	1830	1421	1232	1392	1341
c5315	7314	3327	2028	1698	1343	1225	876	718	687	657	8081	4764	3820	3279	2805	2270	2044	2123	2001	2013
c6288	3448	3101	3078	3129	3251	3216	3340	3177	3043	3039	not reported because c6288 has no low density nodes									
c7552	8855	5511	3463	2861	2503	1722	1558	1359	1264	997	20407	11225	7155	5904	4677	4037	3867	3430	2646	2682

B.4. Execution times on larger circuits

The final issue investigated is the simulator's execution time when processing larger circuits. For the technique to gain wide acceptability, it must have reasonable execution times on larger circuits. The circuits used in this section are the largest ones presented at ISCAS in 1989 [5].

Circuits were first simulated with high $\eta_{\min}$. This provided a rough estimate of each circuit's transition density distribution. The simulation was then rerun with $\eta_{\min}$ chosen to classify under 20% of the nodes as low-density nodes while providing reasonable execution times. The number of gates, execution times, and percentage of low-density nodes are shown for each circuit in Table V. Considering the high accuracy level (5% error at 95% confidence), the execution times are reasonable, especially for the more common class of synchronous circuits, and indicate that this approach is applicable to large circuits.

TABLE V

EXECUTION TIMES IN CPU SECONDS, ON A SUN SPARC ELC

circuit	#gates	synchronous mode		asynchronous mode	
		%low-D nodes	cpu time	%low-D nodes	cpu time
s9234.1	5597	19.6	37.6 min	18.8	1.8 h
s13207.1	7951	18.8	31.9 min	19.2	2.7 h
s15850.1	9772	17.2	45.5 min	17.4	1.7 h
s35932	16065	8.0	1.4 h	10.2	7.4 h
s38584.1	19253	18.1	1.9 h	16.4	7.4 h
s38417	22179	15.0	2.1 h	19.7	7.3 h

IV. CONCLUSIONS

This paper describes a statistical estimation technique, implemented in the program MED, which estimates individual node transition densities with user-specified accuracy and confidence. It uses a threshold $\eta_{\min}$ to classify nodes as either regular- or low-density nodes. Regular-density nodes, $\bar{n} \geq \eta_{\min}$, have transition density values certified to be within a user-specified *percentage* error. Low-density nodes, $\bar{n} < \eta_{\min}$, have transition density values with *absolute* error bounds.

Data were gathered to verify that both regular- and low-density node transition density values are within the stated error bounds. Trials were run with 95% confidence and 5% error tolerance. It was found that well over 95% of regular node transition density values have less than 5% error. This occurs because many of the nodes converge quickly and are subsequently oversampled. Low-density nodes also performed well. Well over 95% of low-density node transition density values have less than the specified absolute error.

Data were also gathered to investigate the variation of execution time with $\eta_{\min}$. In most cases, it was found that the execution time for circuits falls dramatically as $\eta_{\min}$ rises. This occurs because the lowest density nodes typically converge last.

Finally, data were taken for execution times on large circuits. MED required reasonable execution times for large circuits when under 20% of nodes are classified as low-density.

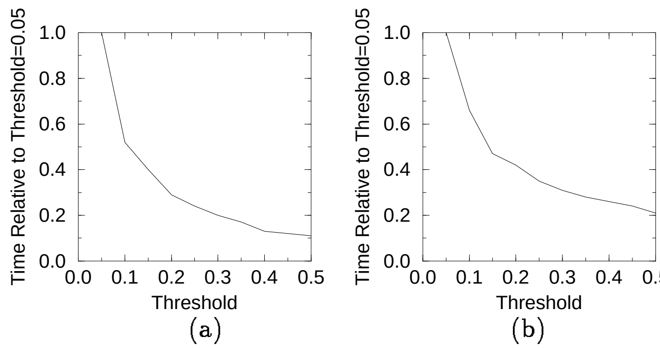

Fig. 2. Typical reduction in execution times with increasing $\eta_{\min}$ for (a) synchronous and (b) asynchronous mode.

ACKNOWLEDGMENT

Discussions with Dr. Richard Burch and Dr. Ping Yang, both of Texas Instruments, provided the motivation for this work. Their contribution is gratefully acknowledged.

REFERENCES

[1] F. Najm, "Transition density: A new measure of activity in digital circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 310–323, Feb. 1993.

[2] R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte Carlo approach for power estimation," *IEEE Transactions on VLSI Systems*, vol. 1, no. 1, pp. 63–71, March 1993.

[3] I. Miller and J. Freund, *Probability and Statistics for Engineers*, 3rd edition. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1985.

[4] F. Brglez, P. Pownall, and R. Hum, "Accelerated ATPG and fault grading via testability analysis," *IEEE International Symposium on Circuits and Systems*, pp. 695–698, June 1985.

[5] F. Brglez, D. Bryan and K. Kozminski, "Combinational profiles of sequential benchmark circuits," *IEEE International Symposium on Circuits and Systems*, pp. 1929–1934, May 1989.