Towards a High-Level Power Estimation Capability†

Farid N. Najm

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

Abstract – We will present a power estimation technique for digital integrated circuits that operates at the register transfer level (RTL). Such a high-level power estimation capability is required in order to provide early warning of any power problems, before the circuit-level design has been specified. Our estimator is based on the use of entropy as a measure of the average activity to be expected in the final implementation of a circuit, given only its Boolean functional description. This technique has been implemented, and empirical results will be presented that demonstrate the feasibility and utility of this approach.

I. Introduction

The high device count and operating frequency of modern integrated circuits has led to unacceptably high levels of chip power consumption. For example, the PowerPC microprocessor from Motorola consumes 8.5 Watts, the Pentium chip from Intel consumes 16 Watts, and DEC's Alpha chip consumes 30 Watts. Due to limited battery life, high power consumption is a major problem in the design of portable or mobile electronics. Even in line-powered equipment, such high power levels require expensive packages and heat-sinks. Thus, there is a need for CAD tools to help with the power management problem.

In order to avoid costly redesign steps, power estimation tools are required that can assess the power dissipation *early* in the design process, before the final circuit-level design has been specified. This allows the designer to explore design trade-offs at a higher level of abstraction than was previously possible, reducing design time and cost.

† This work was supported in part by Intel Corp., Santa Clara, CA.

While several approaches have been proposed for schematic-level power estimation (see [1] for a recent survey), there has been little work on power estimation for general logic circuits at higher levels of abstraction, such as when the circuit is represented only by Boolean equations.

We propose that a way of providing this capability is to make use of the concept of *computational work*, based on the use of *entropy* from information theory. This concept was introduced in the early 70s, as researchers were looking for a measure of the area complexity of a computational process (computer program). It was felt that, by somehow measuring the computational work being performed, one should be able to predict the *area cost* of an implementation. While this sounds reasonable, it turned out to be very difficult to quantify computational work. In 1972, Hellerman [2] proposed the use of *entropy* as a measure of computational work. Entropy will be discussed at length in the next section.

These efforts were mostly unsuccessful [3] for a general computational process, but were reasonably successful [4–8] in the limited context of a combinational logic circuit implementing a Boolean function. Thus, it seems plausible to apply these concepts to perform power estimation of a combinational circuit at a point in the design process where only the Boolean functionality of the circuit, but not its gate-level implementation, is known. The circuit representation at this level of abstraction is usually called a (structural) register-transfer-level (RTL) description. In this description, the circuit is described in terms of well-defined flip-flops or latches and other combinational logic blocks, described only by Boolean functions.

In this paper, we will present a technique for estimating the average switching frequency inside a combinational circuit, given only its input/output Boolean functional description. This represents a first step towards a high-level power estimation capability. The technique is based on properties of the entropy function and a few simplifying assumptions and approximations whose validity will be demonstrated with empirical results.

ACM/IEEE International Symposium on Low Power Design, 1995.

II. Entropy in Logic Circuits

Entropy is a characterization of a random variable or a random process. It is used in information theory [9] as a measure of information-carrying capacity. If x is a random Boolean variable with probability p of being high, i.e., $\mathcal{P}\{x = 1\} = p$, then the entropy of x is defined as:

$$H(x) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{(1-p)} \tag{1}$$

where $\log_2$ is the logarithm to base 2. A plot of $H(x)$ is shown in Fig. 1.

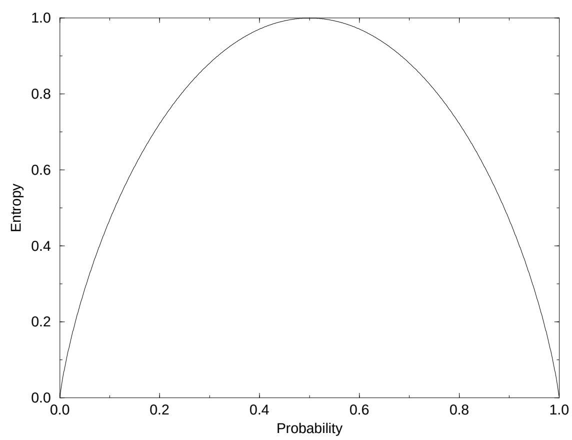

Fig. 1. The entropy of a Boolean variable.

The function $H(x)$ has a maximum value of 1 at $p = 0.5$. Intuitively, if a signal has $p = 0.5$ then it can make the maximum number of transitions and can carry the most information. In general, if a discrete variable can take n different values then its entropy is:

$$H(x) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} \tag{2}$$

where p_i is the probability that x takes the ith value x_i.

Thus every Boolean variable (or vector) has an associated entropy function, whose value is determined by the probability value assigned to the variable (or vector). Let $Y = f(X)$ be a Boolean function where X is a Boolean vector with n bits and Y is a Boolean vector with m bits, i.e., $f(\cdot)$ can be implemented by an n-input m-output logic circuit. Then X can take 2^n values and the *input entropy* of $f(\cdot)$ is:

$$H(X) = \sum_{i=1}^{2^n} p_i \log_2 \frac{1}{p_i} \tag{3}$$

And Y can take 2^m values and the *output entropy* of $f(\cdot)$ is:

$$H(Y) = \sum_{i=1}^{2^m} p_i \log_2 \frac{1}{p_i} \tag{4}$$

With $Y = f(X)$ it can be shown (see [9], page 43) that $H(Y) \le H(X)$, so that the entropy at the output of a combinational circuit is always less than at its input.

Previously, the entropy associated with a Boolean function has been used to predict the *silicon area* required to implement that function, without knowing its schematic-level implementation. Given input probabilities of 0.5, the *output entropy* of a Boolean function has been used to predict the area of its *average minimized implementation*, according to:

$$\mathcal{A} \propto \frac{2^n}{n} H(Y) \tag{5}$$

This was shown to be theoretically valid in the limit (as $n \to \infty$) [5]. For small circuits ($n \le 10$), it was empirically observed [8] that $2^n H(Y)$ provides a good measure of area.

III. Power Estimation

We restrict ourselves to the commonly used static CMOS technology. Consider a combinational logic circuit, composed of N logic gates, whose gate output nodes are denoted x_i, $i = 1, 2, \ldots, N$. If $D(x_i)$ is the transition density [10] of node x_i (average number of logic transitions per second), then the average power consumed by the circuit is:

$$P_{avg} = \frac{1}{2} V_{dd}^2 \sum_{i=1}^{N} C_i D(x_i) \tag{6}$$

where C_i is the total capacitance at node i. This expression accounts only for the capacitive charging/discharging component of power, and not for the so-called short-circuit power which is known to be only around 10% of the total power in well-designed circuits. The transition density is a measure of circuit switching *activity*. We will be using the terms "density" and "activity" interchangeably.

Given that the internal details of the logic circuit are not known in a high-level representation, then a few approximations seem inevitable for high-level power estimation. The impact and utility of these approximations will be demonstrated through empirical results in section IV. We start with:

$$P_{avg} \propto \sum_{i=1}^{N} C_i D(x_i) \approx \mathcal{D} \sum_{i=1}^{N} C_i \tag{7}$$

where $\mathcal{D}$ is the *average node transition density*, defined by:

$$\mathcal{D} = \frac{1}{N} \sum_{i=1}^{N} D(x_i) \tag{8}$$

so that:

$$P_{avg} \propto \mathcal{A} \times \mathcal{D} \tag{9}$$

where $\mathcal{A}$ is an estimate of the *circuit area* that is representative of the capacitance $\sum_{i=1}^{N} C_i$.

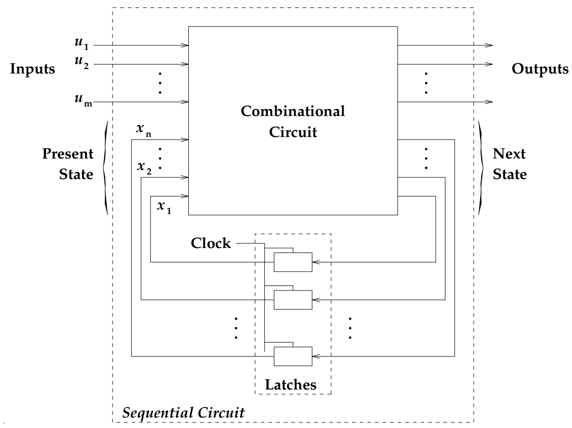

Fig. 2. A general synchronous sequential circuit.

It also seems inevitable that both $\mathcal{D}$ and $\mathcal{A}$ will be estimated only from knowledge of the input/output behavior. As mentioned above, some studies have shown that the area $\mathcal{A}$ is related to entropy. The objective of this paper is to illustrate the relationship between the average density $\mathcal{D}$ and entropy. We will assume throughout that we are dealing with a combinational circuit block that is part of a synchronous sequential circuit, as shown in Fig. 2. If both area and average density are successfully related to entropy, then a viable high-level power estimation methodology would be as follows:

1. Run a structural RTL simulation of the sequential circuit to measure the input/output entropies of the combinational block.
2. From the input/output entropies, estimate $\mathcal{D}$, $\mathcal{A}$, and P_{avg} for the combinational block.
3. Combine with latch and clock power to get the total average power.

In the two sub-sections below, we discuss the estimation of entropy from an RTL simulation trace (step 1), and the estimation of average density from entropy (missing part of step 2). Step 3 is easy, given the clock frequency and the results of steps 1 and 2.

A. Entropy from RTL Trace

If $X = [x_1, x_2, \ldots, x_n]$ is a Boolean vector (say, the *next state* vector or *present state* vector) then it is obviously too expensive to estimate its entropy from the definition:

$$H(X) = \sum_{i=1}^{2^n} p_i \log_2 \frac{1}{p_i} \tag{10}$$

Instead, we can efficiently estimate an upper bound on the entropy based on the fact $H(X) \leq \sum_{i=1}^{n} H(x_i)$, where equality occurs when the signals x_i are independent [9], and where $H(x_i)$ is the entropy of x_i evaluated using the probability of x_i (these probabilities can be computed as in [13, 14], or in [11]). We will therefore assume that the nodes in a cross-section of the circuit are not too correlated, and will make the approximation:

$$H(X) \approx \sum_{i=1}^{n} H(x_i) \tag{11}$$

whose impact has been found to be acceptable, as will be shown by experimental results. In any case, using the upper bound is a conservative approximation, because it never underestimates the entropy and thus errs on the side of higher activity.

B. Average Activity from Entropy

Consider one of the *present state* bit signals, and let p be its signal probability (average fraction of clock cycles in which it is high). If the signal values in two consecutive cycles are assumed independent, then its average activity per clock cycle is $2p(1-p)$ transitions per cycle (the transition density is $2p(1-p)/T_c$ transitions per second, where T_c is the clock period [1]). It so happens that the plot of $4p(1-p)$ is very close to that of the entropy function, as shown in Fig. 3.

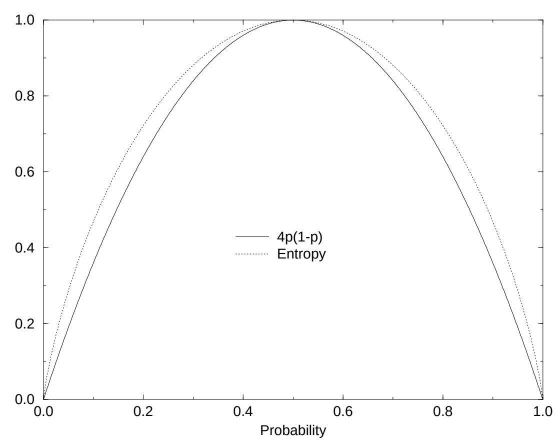

Fig. 3. The relation between activity and entropy.

Thus it makes sense to use entropy as a measure of activity, so that if $\mathcal{H}$ is the average value of $H(x_i)$ over all nodes x_i in the circuit, then (with some approximation):

$$P_{avg} \propto \mathcal{A} \times \mathcal{H} \tag{12}$$

This formulation is useful because it makes it possible to use special properties of the entropy such as "output entropy is always less than input entropy" to help predict the average internal node entropy of a combinational block from its input/output entropy, as follows.

A combinational circuit can always be *levelized* so that its gates are tagged with *level* values that represent their distance from the primary inputs. Thus every gate whose inputs are all primary inputs is said to have level 1. Every other gate whose inputs are either outputs of level 1 gates or are primary inputs is said to have level 2, etc. The levelization algorithm [12] has linear time complexity and is standard in most logic/timing simulation systems.

The largest level number K used in levelizing a circuit is called the circuit *depth*. For every level $i = 1, 2, \ldots, K$, the output nodes of the gates at level i are said to form a *cross-section* of the circuit. We define $H(i)$ to be the entropy of the Boolean vector consisting of the nodes at the cross-section at level i. Thus $H(K)$ is the entropy of the vector of primary output nodes (next state vector + output vector, in the case of a sequential circuit), denoted by H_o. We define $H(0)$ to be the entropy of the vector of primary input nodes (present state vector + primary inputs vector, in the case of a sequential circuit), denoted by H_i. As pointed out above, we always have $H_o \leq H_i$ and, in general, $H(j) \leq H_i$, for $j = 1, 2, \ldots, K$.

It has been empirically observed [8] that output entropy of a circuit decreases *quadratically* with circuit depth. We therefore assume that the entropy of a cross-section decreases quadratically with level number, so that the entropy at a cross-section at level j, $H(j)$, may be written as:

$$H(j) = (H_i - H_o)\left(1 - \frac{j}{K}\right)^2 + H_o \qquad (13)$$

where K is the number of levels in the circuit. We will use this to derive an expression for the *average node entropy* $\mathcal{H}$, starting with the approximation:

$$\mathcal{H} = \frac{1}{N}\sum_{i=1}^{N} H(x_i) \approx \frac{1}{N}\sum_{j=1}^{K} H(j) \qquad (14)$$

which is based on the conservative approximation (11) that the entropy of a cross-section is approximately equal to its upper bound (the sum of its constituent node entropies). And, as usual, N is the total number of gates in the circuit.

Let $W(i)$ be the number of nodes in cross-section i, which we will call the circuit *width* at that cross-section. If m is the number of primary output nodes, then $W(K) \leq m$. We define $W(0)$ to be the number of primary input nodes, also denoted by n. Let $\mathcal{W}$ be the *average width* of the circuit, defined as:

$$\mathcal{W} = \frac{1}{K}\sum_{i=1}^{K} W(i) = \frac{N}{K} \qquad (15)$$

From the above, it follows that:

$$\begin{aligned}
\mathcal{HW}K &= \sum_{j=1}^{K} H(j) \\
&= KH_o + (H_i - H_o)\sum_{j=1}^{K}\left(1 - \frac{j}{K}\right)^2 \\
&= KH_o + (H_i - H_o)\frac{1}{K^2}\sum_{j=1}^{K}(K-j)^2 \\
&= KH_o + (H_i - H_o)\frac{1}{K^2}\sum_{k=1}^{K-1} k^2 \\
&= KH_o + (H_i - H_o)\frac{(K-1)K(2K-1)}{6K^2} \\
&\approx KH_o + \frac{K}{3}(H_i - H_o)
\end{aligned} \qquad (16)$$

where the last approximation is conservative because actually $(K-1)(2K-1) \leq 2K^2$, so that, again, we err on the side of higher activity. This leads to:

$$\mathcal{HW} \approx \frac{H_i + 2H_o}{3} \qquad (17)$$

which does not depend on circuit depth (a must, so as to be applicable to a high-level representation). If we further approximate the average circuit width by $\mathcal{W} \approx (n+m)/2$, which is approximately the average of the widths at the input and output sides, then we arrive at the simple final expression:

$$\mathcal{H} \approx \frac{2/3}{n+m}(H_i + 2H_o) \qquad (18)$$

which depends only on the input and output entropies and on the input and output node counts, all of which are obtainable from a high-level representation.

In spite of the approximations made above, we have found that the resulting simple expression for $\mathcal{H}$, (18), works quite well for a broad range of circuits. The empirical results presented in the next section will be based on this expression.

IV. Empirical Results

As a first step towards a high-level power estimation capability, we have implemented a technique for estimating the average node activity of a combinational circuit, based on the average entropy measure (18). To use this technique, we estimate H_i and H_o from their upper bounds, according to (11) and then use (18) to predict the average activity. We tested the technique on isolated combinational circuit blocks whose input probabilities are user-specified. Normally, the input probabilities would be obtained from an examination of the behavior of the sequential circuit as in the established techniques [13, 14] or [11]. The input probabilities are enough to compute H_i, but H_o depends on the output probabilities. These can be computed using BDDs as explained in [10], but this can be memory intensive. Instead, we compute them using a Monté Carlo approach, as described in [15].

In order to assess the accuracy of the technique, we need an accurate measure of average node activity, obtained from a schematic-level view of the same circuit. This can be obtained by first finding the transition density at every node and then averaging the results. Accurate transition density values were obtained by simulation in two ways, depending on the timing model chosen:

1. Using a *zero-delay* timing model: In this case, one is interested only in the final steady state node values in a clock cycle, and any additional toggles due to unequal delay paths are ignored. In this case, also, the density is easily obtained from the signal probability [1] as $D(x) = 2p(1-p)/T_c$, and the probabilities were obtained using the technique in [15].
2. Using a *general-delay* timing model: In this case, the delays are obtained from a gate library and an event driven simulation is performed as in [16].

The delay model did not enter into the derivation of (18), as is probably to be expected in a high-level model. Therefore, in order to check the impact of the approximations made in the derivation, it is important to check the accuracy of (18) against the zero-delay simulation results.

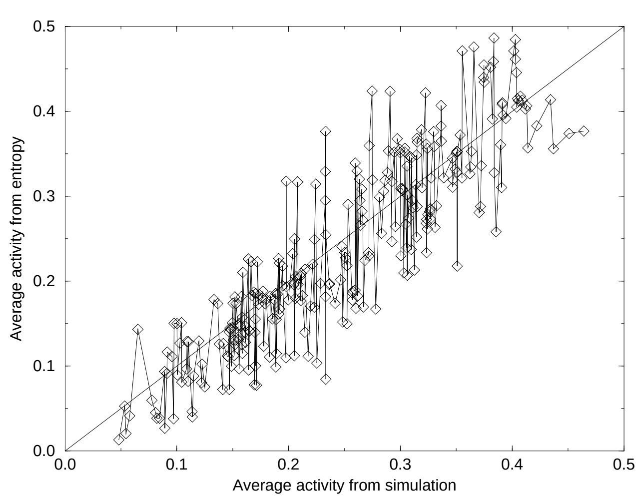

Fig. 4. Comparison with zero-delay results.

The results of testing against the zero-delay analysis results are shown in Fig. 4 for 56 different circuits, with sizes ranging from 100 to about 22,000 gates, and with input probabilities ranging from 0.1 to 0.9. These circuits include all the ISCAS-85 and ISCAS-89 circuits. As shown in Fig. 3, the average entropy should correlate well with twice the average activity per clock cycle. Thus the "activity from simulation" shown on the horizontal axis in Fig. 4 is actually normalized to give the average value of $4p(1-p)$ for each circuit. The agreement is quite good, with an error of less than 0.09, with 90% confidence. We consider this to be strong indication that the technique is feasible and constitutes a reasonable approach to high-level power estimation. The approach is also very fast. Our implementation, which includes reading the circuit, estimating the output entropy, and evaluating (18), requires only 14 cpu seconds for a 20,000 gate circuit (on a SUN sparc-10).

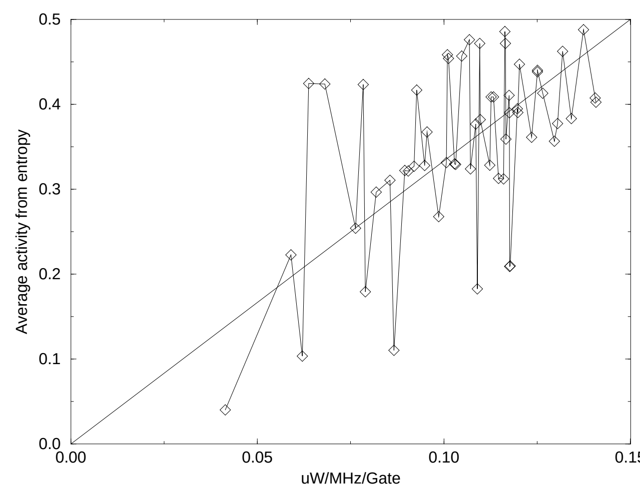

Fig. 5. Comparison with zero-delay power/area results.

The effect of capacitance is not included in the data shown Fig. 4 (only activity values were measured, and not activity $\times$ capacitance). Therefore, before moving to the case of the general delay timing model, we tested the impact of the approximation (7) which is equivalent to an independence assumption between the node capacitance and node density distributions. To do this, we checked if the average entropy correlates with the power per unit area, according to $\mathcal{D} \propto P/\mathcal{A}$. We used gate count as a measure of area, and estimated power using a zero-delay timing model, accounting for fanout capacitance. Average entropy is compared to power per unit area, in units of μW/MHz/gate, in Fig. 5. The results shown are for the same circuits used in Fig. 4, but only for an input probability value of 0.5. The results show slightly more spread than Fig. 4, due to the effect of the node capacitance distribution.

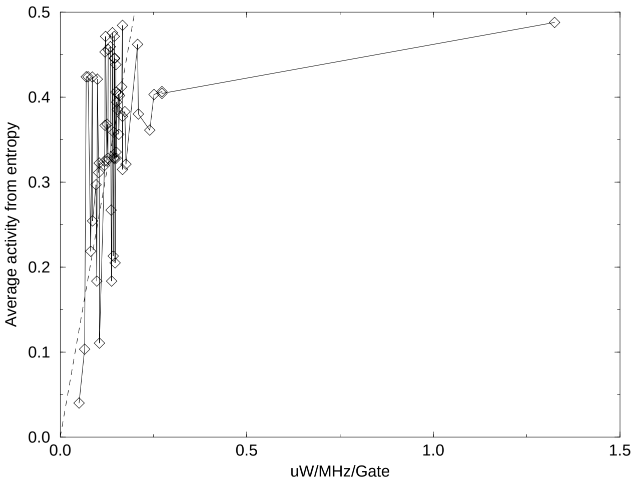

Fig. 6. Comparison with general-delay results.

Finally, we measured the power under a general timing model. The power in some circuits increases appreciably due to multiple transitions/cycle. We compare the average activity measured from entropy to the power/area, in μW/MHz/gate, as shown in Fig. 6. For one circuit (iscas-85:c6288), the deviation was very large, as shown by the point at the far right in the figure. Hence more work is needed to predict situations like this. Furthermore, the comparisons for the other circuits are not as good as before and show increased spread, as shown in Fig. 7.

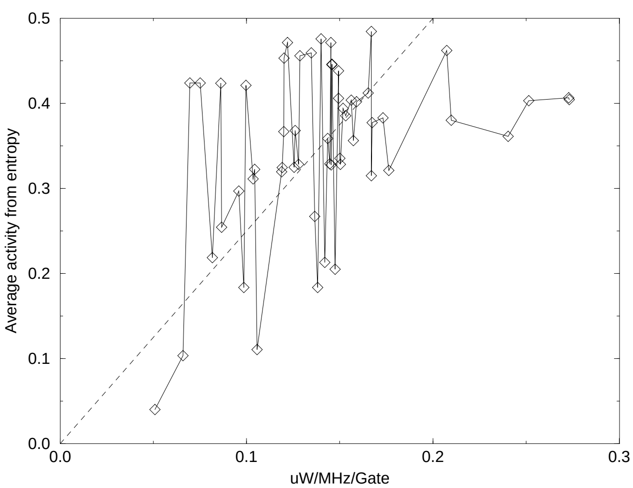

Fig. 7. Comparison with general-delay results.

V. Conclusions

There is a need for high-level power estimation, and the RTL level seems the reasonable place to start. We proposed to use computational work, based on entropy, as a high-level measure of power. Preliminary investigation shows that entropy is a viable measure of circuit activity, but needs improvement to account for general delay and capacitance distribution.

References

[1] F. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Transactions on VLSI Systems*, pp. 446–455, Dec. 1994.

[2] L. Hellerman, "A measure of computational work," *IEEE Transactions on Computers*, vol. C–21, no. 5, pp. 439–446, May 1972.

[3] K. Mase, "Comments on "A measure of computation work" and "Logical network cost and entropy," *IEEE Transactions on Computers*, vol. C–27, no. 1, pp. 94–95, January 1978.

[4] R. W. Cook, and M. J. Flynn, "Logical network cost and entropy," *IEEE Transactions on Computers*, vol. C-22, no. 9, pp. 823–826, September 1973.

[5] N. Pippenger, "Information theory and the complexity of Boolean functions," *Mathematical Systems Theory*, vol. 10, New York: Springer-Verlag Inc., pp. 129–167, 1977.

[6] J. A. Dussault, "A testability measure," *IEEE Digital Semiconductor Test Symposium*, Cherry Hill, NJ, pp. 113-116, Oct–Nov. 1978.

[7] V. D. Agrawal, "Information theory in digital testing - a new approach to functional test pattern generation," *IEEE International Conference on Circuits and Computers*, Port Chester, NY, pp. 928–931, Oct. 1–3, 1980.

[8] K-T Cheng and V. Agrawal, "An entropy measure for the complexity of multi-output Boolean functions," *27th ACM/IEEE Design Automation Conference*, Orlando, FL, pp. 302–305, June 24–28, 1990.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York, NY: John Wiley & Sons, 1991.

[10] F. Najm, "Transition density : a new measure of activity in digital circuits," *IEEE Transactions on Computer-Aided Design*, vol. 12, no. 2, pp. 310–323, February 1993.

[11] F. N. Najm, S. Goel, and I. N. Hajj, "Power Estimation in Sequential Circuits," *ACM/IEEE Design Automation Conference*, 1995.

[12] S. Even, *Graph Algorithms*, Rockville, MD: Computer Science Press, 1979.

[13] J. Monteiro and S. Devadas, "A methodology for efficient estimation of switching activity in sequential logic circuits," *ACM/IEEE 31st Design Automation Conference*, San Diego, CA, pp. 12–17, June 6–10, 1994.

[14] C-Y Tsui, M. Pedram, and A. M. Despain, "Exact and approximate methods for calculating signal and transition probabilities in FSMs," *ACM/IEEE 31st Design Automation Conference*, San Diego, CA, pp. 18–23, June 6–10, 1994.

[15] F. Najm, "Statistical Estimation of the Signal Probability in VLSI Circuits," Coordinated Science Laboratory Report #UILU-ENG-93-2211, April 1993.

[16] M. Xakellis and F. Najm, "Statistical Estimation of the Switching Activity in Digital Circuits," *31st ACM/IEEE Design Automation Conference*, San Diego, CA, pp. 728–733, 1994.