An Entropy Measure for Power Estimation of Boolean Functions †

Chi-Hong Hwang and Allen C.-H. Wu

Department of Computer Science
Tsing Hua University
Hsinchu, Taiwan, 300, Republic of China

Abstract

In this paper, we present a study on the relationship between entropy and the average power consumption of circuits generated from Boolean functions. Based on a general-delay model, an entropy-based formulation for power estimation is derived from a large set of experimental data. The study shows that the entropy measure provides an effective power estimate for single-output and fully-correlated multiple-output functions. The study also shows that if entropy is used as a power measure, the internal structure of a circuit must be considered in order to achieve accurate power estimates for non-correlated multiple-output functions. Experiments on a set of benchmarks demonstrate that combining entropy-based power measures with input-output correlation analyses of logic functions leads to a viable measure for high-level power estimation.

1 Introduction

With the advent in portable computing and high density VLSI circuits, power dissipation has emerged as a principle design consideration in VLSI designs. In the past few years, handful power minimization methods have been reported for low power designs at circuit, layout, logic, behavioral, and architectural levels [1, 2].

Accurate and efficient power estimation methods are indispensable to the design of low power applications. In the absence of a power estimation and analysis tool, designers will have difficulty to make intelligent design decisions during the design process in order to meet the power requirement without a costly redesign process. Hence, a variety of power estimation techniques have been proposed and studied extensively in industry and academia [3, 4].

Most of reported power estimation techniques are focusing on power estimation at circuit and schematic levels. Only very few of them are targeted to the power estimation for designs at higher levels of abstraction such as Boolean functions and RT-level designs. In recent studies, Najm [5] proposed a technique for power estimation of designs at the register-transfer level (RTL). This method uses the entropy as a measure of the average switching activity of the expecting final circuit from its Boolean function. Marculescu et al. [6] proposed two RTL power estimation methods. The first method uses the entropy to predict the average switching activity of the circuit. The second one uses the information energy of the circuit to predict the average switching activity. Both studies demonstrate that entropy is a viable power measure for circuits from their logic functions.

In this paper, we address the problem of the use of entropy to predict the average power consumption of circuits from their Boolean functions. In the previous studies, entropy has been successfully used for solving the problems of area estimation [8, 9, 11] and average switching activity estimation [5, 6]. In this study, we investigate the direct relationship between entropy and the average power consumption of circuits through a series of experiments. An entropy-based formulation for power estimation is derived from the data based on a general-delay model. The usages and limitations of using entropy for high-level power estimation are discussed in the paper.

This paper is organized as follows. Sections 2 and 3 introduce the basic concepts of entropy measures for area and power estimations. Section 4 presents the empirical experiments. Section 5 discusses the experiments using entropy for power estimation. Finally, section 6 gives concluding remarks.

2 Entropy Measures for Area Estimation of Boolean Functions

Entropy has been formulated as an area measure for implementing a Boolean function [7, 8, 9, 10]. In this section, we first describe the entropy of a Boolean function and then the use of entropy to predict circuit areas of Boolean functions.

Given a Boolean function with n inputs and m outputs, there are 2^n possible input vectors and 2^m output vectors. Assume that the input values are uniformly distributed, the probabilistic distribution of the outputs can be described as a random variable. For each output vector O_i, the probability of $f = O_i$ is

$$Pr(f = O_i) = P_i = N_{O_i}/2^n, \qquad (1)$$

where N_{O_i} is the number of times O_i appears in the

†Supported by the National Science Council of R.O.C. under contracts No. NSC-85-2215-E-007-034 and NSC-86-2221-E-007-021

ASP-DAC '97
0-89791-851-7/$5.00

truth table.

The entropy H of a boolean function f is defined as

$$H(f) = \sum_{i=1}^{2^m} (P_i \cdot log_2(1/P_i)). \quad (2)$$

For a single-output function f, the above equation can be reduced to

$$H(f) = P_1 \cdot log_2(1/P_1) + (1-P_1) \cdot log_2(1/(1-P_1)), \quad (3)$$

where P_1 is the fraction of 1's in the output column of the truth table.

Under the presumption of that circuit area is proportional to the number of logic devices used in the computational work, Hellerman [8] proposed the use of entropy as a measure for the computational work of an n-input Boolean function, which is defined as Computational Work $= 2^n H(f)$. In the later work, Pippenger [10] proposed a formula to predict the average amount of logic required for implementing an n-input Boolean function, which is shown as follows:

$$L = (1-d) \cdot K(n) \cdot H(f_1), \quad (4)$$

where L is the literal count, d is the fraction of don't care, and $K(n)$ is a function of the number of inputs.

Cheng and Agrawal [11] conducted an extensive empirical experiment to evaluate formula 4. The results indicate that computational work based on the entropy of the Boolean function has shown a direct relation to the hardware needed to implement the function. In their study, they have determined the values of $K(n)$ $(n = 6-10)$, which also confirms the previous observations [12] of $K(n) = k2^n$ for some constant k. Furthermore, the results also indicate that the average amount of literal count is always given by the computational work formula irrespective of the number of inputs and outputs of the Boolean functions. This demonstrates that entropy is a useful measure for area estimation from Boolean functions.

3 Entropy and Circuit Power

Recently, Najm and Marculescu et al., [5, 6] proposed two entropy-based power estimation approaches for logic/RT-level designs. Both studies demonstrate that entropy is a viable measure for power estimation of a circuit from its logic function. In this section, we will introduce the basic concept of the use of entropy for power estimation.

As stated in [5], the average power P_{avg} consumed by a circuit is

$$P_{avg} = 1/2 V_{dd}^2 \sum_{i=1} C_i \cdot D(x_i), \quad (5)$$

where C_i is the total capacitance at node i and $D(x_i)$ is the transition density of node x_i.

Hence, the average power consumed by a circuit is proportional to the sum of the product of transition density and the capacitance of all nodes in the circuit as

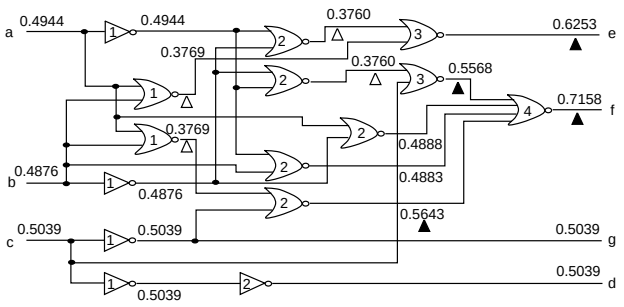

Figure 1: An example.

$$P_{avg} \propto \sum_{i=1}^{N} C_i \cdot D(x_i) \approx D \sum_{i=1}^{N} C_i, \quad (6)$$

where D is the average node transition density.

Since the total capacitance of all nodes is proportional to the circuit area A, the average power consumption of a circuit can be expressed as

$$P_{avg} \propto A \cdot D. \quad (7)$$

As mentioned in the previous section, area A can be estimated using a formula based on the entropy of a Boolean function (i.e., formula 4). Thus, if the average transition density of a circuit is also related to entropy, then the average power consumed by a circuit can be estimated directly from the entropy of the circuit. As a result, the main objective in [5] is to establish the relationship between the average transition density and the entropy of a circuit.

Under the assumptions of that the average transition density of a circuit is proportional to the average entropy and the output entropy of a circuit is a decreasing function with the circuit depth (i.e., output entropy is always less than input entropy), Najm [5] proposed an approximation of the average value of $H(X)$ over all nodes x_i in the circuit, which depends only on the input and output entropies and the input and output node counts as

$$H \approx (2/3)/(n+m)(H_i + 2H_o), \quad (8)$$

where n and m are the number of inputs and outputs, H_i and H_o are the input and output entropies, respectively.

The results in [5] show that using the zero-delay timing model the estimated average transition densities obtained using formula 8 are well correlated to the one generated by simulation. However, this model may not work well on a general-delay timing model because transition density does not react as a uniform decreasing function with circuit depth on a general-delay model. For example, Figure 1 shows a circuit in which the output entropies are not always decreased with the circuit depth. In Figure 1, the solid triangles and empty triangles indicate the transition densities which increase and decrease with the circuit depth, respectively.

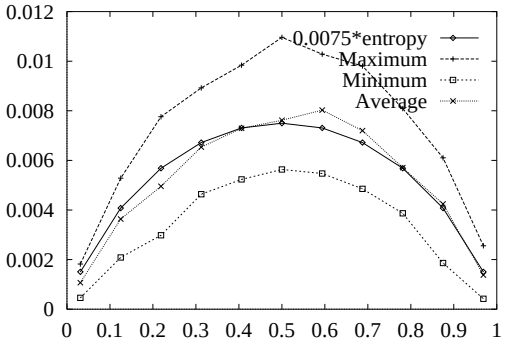

Figure 2: Functions with 6-input and single-output.

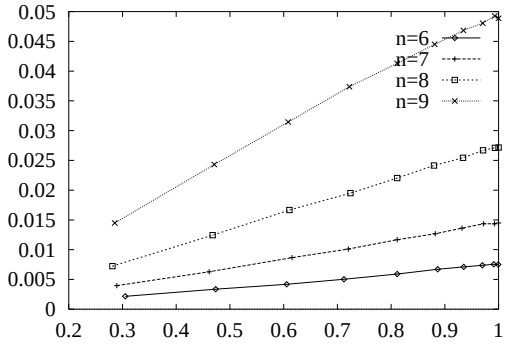

Figure 3: Fully specified n-input single-output functions.(n=6,7,8,and 9)

4 An Empirical Study

In this section, we will present an empirical study to investigate the relationship between the power and entropy of a Boolean function based on a general-delay timing model. We first used the approach described in [11] to conduct a series of empirical experiments. We assumed that the signal probability of each input is 1/2 and the transition density is 1/2 of the clock frequency. For a given $P1$ (probability of $f = 1$), we randomly generated 100 functions. These functions were then minimized using the MIS [13] system (using the standard script). The circuits were mapped with the *mcnc.genlib* library. Finally, we used the MED system [14], which is a Monte-Carlo based power estimator, to estimate the power consumptions of the generated circuits.

Figure 2 shows the relationship between the power consumption and $P1$ for 6-input and single-output functions. The points on the four curves are the maximum, average, and minimum power consumptions (mW/MHz) obtained from 100 functions, and the normalized entropy function to fit the measured power consumption. In addition, the relation graph of the average power consumptions versus the entropies for 6, 7, 8, and 9-input functions is shown in Figure 3, in which each point represents an average value over 200 randomly generated functions. The results indicate a near-linear relationship between power and entropy.

From the data, we found that the power consumption of a Boolean function can be formulated as follows:

$$P_{avg} = K_p(n) \cdot H(f), \tag{9}$$

which is same as the area formulation (formula 4) proposed by Pippenger [10] on fully specified functions.

Table 1: Values of $K_p(n)$

n	$K_p(n)$	$K_p(n)/K_p(n-1)$
6	0.00750	-
7	0.01494	1.99
8	0.02716	1.82
9	0.04888	1.82

From the analysis of measured slope for functions with different number of inputs (Table 1), we found that $K_p(n)/K_p(n-1) \approx 2$ which is same to the observations [12, 11] on the relationship between the area and entropy of a Boolean function. Hence, we can express the relationship between the power consumption and entropy of a Boolean function as below:

$$P_{avg} = k_p \cdot 2^n \cdot H(f), \tag{10}$$

for some constant k_p.

We have also conducted experiments on a set of multiple-output functions. As described earlier, we randomly generated a large number of functions (1000 each for (6-input, 2-output), (6-input, 3-output), (6-input, 4-output (Figure 4)), (7-input, 2-output (Figure 5)), (8-input, 4-output (Figure 6)), and 100 for (9-input, 7-output (Figure 7))). We assumed that the outputs of the randomly generated functions are correlated to each input. From the data, we also observed the same results as indicated in [11] such that the average power consumed by a circuit is always given by formula 10 irrespective of the number of inputs and outputs.

Furthermore, we have conducted the same set of experiments by applying different mapping methods. The 100 generated functions were minimized using the MIS system with the standard script and then mapped using default, area and delay options (default:map -W -p, area:map -W -m O -AF -p, and delay:map -W -n 1 -AFG -p) with the *mcnc.genlib* library. Figures 8 and 9 show the the average power and entropy relationship with different technology mapping options on 6-input single-output and 9-input single-output functions, respectively. The results also conform to the previous derived formula (formula 10) with a varying constant k_p. One interesting observation from the data is that circuits generated using the mapping with the minimum-area option do not produce the minimum-power designs.

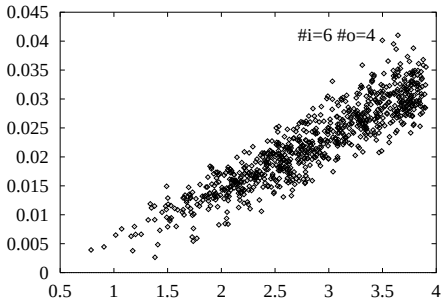

Figure 4: Functions with 6-input and 4-output.

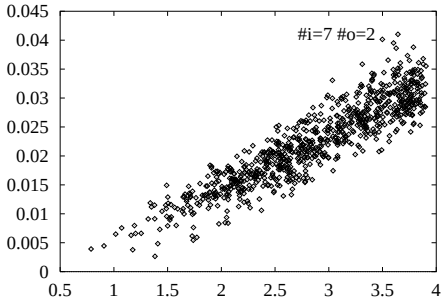

Figure 5: Functions with 7-input and 2-output.

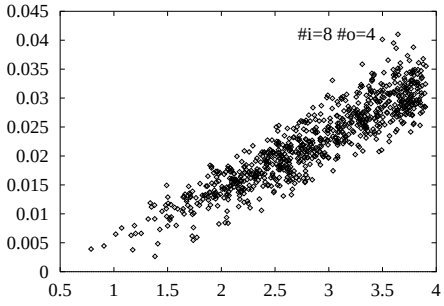

Figure 6: Functions with 8-input and 4-output.

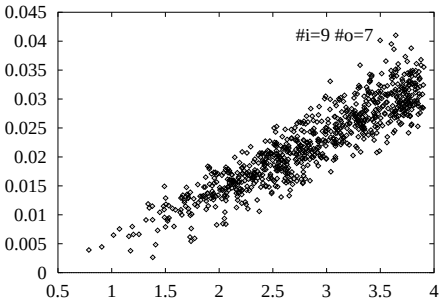

Figure 7: Functions with 9-input and 7-output.

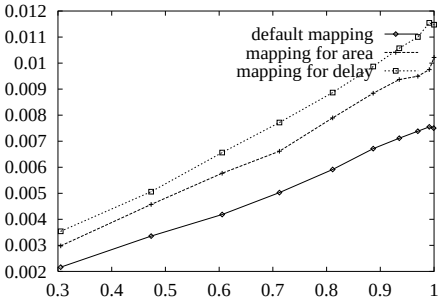

Figure 8: Applying different mapping options on functions with 6-input and single-output.

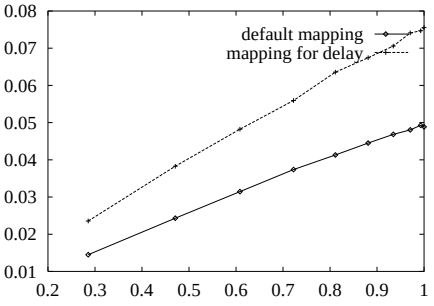

Figure 9: Applying different mapping options functions with 9-input and single-output.

5 Experiments

In this section, we conducted experiments using entropy as a power measure. First, we applied formula 10 to estimate the power consumptions of a set of benchmarks. The results are shown on the P_{avg} column in Table 2. The results show that the power measures using formula 10 are overestimated by a large margin on the multiple-output examples compared to that generated by simulation (P_{avg} v.s. MED [14]).

The main contribution to such overestimations is described as follows. Because the entropy of a random variable depends solely on the probabilistic distribution, using entropy as a power measure which only considers the probabilistic distribution of the Boolean value but not the structure of its generated circuit. For single-output functions with the same entropy, they all have the same number of $1's$ in the truth table. Hence, we will obtain the same power consumption for all these functions when entropy is used as a power measure. However, because of the different distribution of 1's location within the functions, the synthesized circuits from these functions may vary in a wide range so as the power consumed by the generated circuits. This explains the deviation of the power consumptions under the same entropy, as shown in Figure 2.

Table 2 Power estimation for benchmarks

Ckt.	#I/Os	MED	$H(f)$	P_{avg}	P_{avg}*
9symm1	9/1	0.0472	0.869	0.042	0.042
majority	5/1	0.0022	0.992	0.0037	0.0037
cm138a	6/8	0.0022	0.919	0.0068	0.0027
cm42a	4/10	0.0044	3.031	0.0056	0.0043
b1	3/4	0.0022	2.5	0.002	0.0012
cm82a	5/3	0.0055	2.479	0.0092	0.0053
f51m	8/8	0.0412	8	0.2173	0.0267
x2	10/7	0.0066	2.958	0.2892	0.052
z4ml	7/4	0.0095	3.730	0.0557	0.018
alu2	10/6	0.0823	4.698	0.4592	0.0713

This problem is getting even worse when using entropy for power estimation of multiple-output functions. For example, consider the following worst scenario, given an m-input m-output identity function, the entropy of the function will be m. For this particular function, a very high power estimate will be obtained when entropy is used as a power measure. However, this function can be implemented as a circuit of simply connecting the inputs directly to the outputs, which should consume very little power. This demonstrates that if entropy is used as a power measure, the internal structure of a circuit must be considered in order to achieve a meaningful power estimate.

Let's first investigate why the entropy-based power estimates of multiple-output functions are overly estimated. Recall that in our experiments on multiple-output functions, we assume that the outputs of the randomly generated functions are fully correlated to each input. For example, for an 8-input 2-output function, the two outputs are fully correlated to the 8 inputs. Using formula 10, the power estimate of an 8-input 2-output function is equivalent to the sum of the power estimates of two 8-input single-output functions. However, in most of benchmarking examples, their outputs are not always fully correlated to each input. For example, for the same 8-input 2-output function, if the two outputs only correlate to 7 inputs each, then the power estimate of this function is equivalent to the sum of the power estimates of two 7-input single-output functions. Using formula 10, the power estimate of an 8-input 2-output is larger than the sum of the power estimates of two 7-input single-output functions. This indicates that we have to take into account the correlation between inputs and outputs of logic functions in order to obtain accurate power estimates for multiple-output functions.

In this study, we used a partitioning-based method to perform power estimation of multiple-output functions. The main idea of this approach is to decompose a multiple-output function into a set of single-output functions. We first compute the power consumption for each decomposed single-output function using formula 10. Then, we determine the correlation (called *input-sharing factors*) between inputs and outputs. The power consumption of the multiple-output function is computed as the difference between the sum of power measures of the decomposed single-output functions and the power contributed by the input-sharing factors. The partitioning-based power estimation method for multiple-output functions is listed as follows:

1. Decompose the multiple-output function into a set of single-output functions.
2. Compute the entropy and power for each single-output function.
3. Compute the input-sharing factor for each input.
4. Recompute the power by taking into account the input-sharing factor for each single-output function.
5. Sum up the power measures of all the decomposed single-output functions.

We will use the following example to explain the proposed method. Assume there is a multiple-output function with two outputs o_1 and o_2 and three inputs i_1, i_2 and i_3 in which o_1 is correlated to i_1 and i_2 while o_2 is correlated to i_2 and i_3. We can decompose the function into two single-output functions of $o_1 = f(i_1, i_2)$ and $o_2 = f(i_2, i_3)$. We first use formula 10 to compute the power for these two single-output functions. We then compute the input-sharing factor for each input. For example, since functions of o_1 and o_2 are only correlated to inputs i_1 and i_3, respectively, the input-sharing factors of i_1 and i_3 are one. On the other hand, both of o_1 and o_2 are correlated to i_2. In other words, i_2 is sharing by the functions of o_1 and o_2. Hence, the input sharing factor of i_2 is $1/2$. As a result, the total number of effective inputs of o_1 and o_2 is $1 + 0.5 = 1.5$. Furthermore, the total power factor PF_i of a single-output function i is calculated as $PF_i = (m_i - Eff_i)/m_i$, where m_i is the number of the inputs of function i and Eff_i is the number of effective inputs of function i. For example, for both of o_1 and o_2, the input-sharing factor is $(2 - 1.5)/2 = 1/4$. After obtaining the input-sharing factor for each single-output function, we can

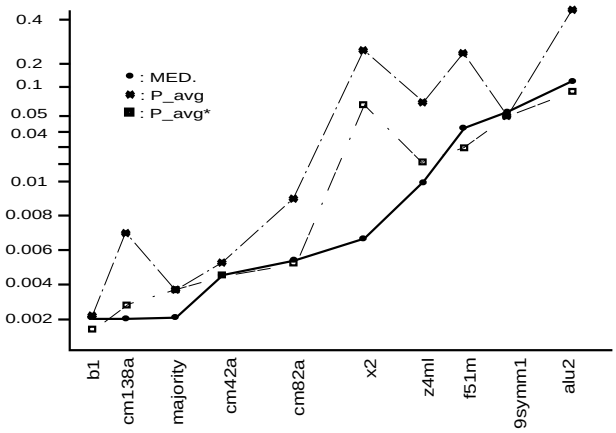

Figure 10: Fidelity analysis of power estimates.

recompute the power for each function by deducting the power contributed by the input-sharing effect. We assume that the power contributed by the input-sharing is proportional to the square of the input-sharing factor. For example, the power of o_1 (P_{o_1}) is recomputed as $P_{o_1} = P_{o_1} \times (1 - (1/4)^2)$. Finally, the power of the multiple-output function is calculated as the sum of power measures of all the decomposed single-output functions.

Using the described method, we re-estimated the power consumptions which are shown on the $P_{avg}*$ column in Table 2. The results show a large degree of improvement on the power estimates. However, the results also indicate that unless we can successfully and accurately predict the structure of a circuit from its Boolean function, using the entropy method to obtain accurate power estimates may not be plausible.

An entropy-based power estimation method may not achieve accurate power estimates for high-level applications. It still poses another useful application to support designers for their design decisions. For example, given two functions, a designer may like to know which one will consume less power to support their design tradeoff decisions. In this case, the "fidelity" of the estimates is more important than the "accuracy". Fidelity is a crucial factor in the quality measure that indicates the degree of the estimated results correspond to the actual results. In other words, fidelity is the deviation from the average error over all design points. If the error over all design points is always of the same magnitude then fidelity is high. We studied the "fidelity" of our power estimates P_{avg} and $P_{avg}*$ to the simulated power estimates (MED). Figure 10 illustrates the fidelity analysis of the power estimates. The results show that a coherent relationship exists between $P_{avg}*$ and the simulated estimates (MED) for all examples except for $x2$. On the other hand, P_{avg} did not provide high fidelity power estimates compared to the simulated power estimates.

6 Conclusions

We have presented a statistical study to derive an entropy formulation for power estimation of Boolean functions. We found that the power consumption of a Boolean function can be formulated as the same formulation proposed by Pippenger for area estimation with a different constant value (k_p). The study showed that the entropy measure provides an effective power estimate for single-output and fully-correlated multiple-output functions. The study also showed that if entropy is used as a power measure, the internal structure of a circuit must be considered in order to achieve accurate power estimates for non-correlated multiple-output functions. We have proposed a partitioning-based power method for multiple-output functions. Experiments on a set of benchmarks demonstrated that combining entropy-based power measures with input-output correlation analyses of logic functions leads to a viable measure for high-level power estimation. Further study in this area is necessary in order to achieve accurate power estimates for high-level applications.

References

[1] S. Devadas and S. Malik, "A Survey of Uptimization Techniques Targeting Low Power VLSI Circuits," in *Proc. of 32nd Design Automation Conf.*, pp. 242-247, 1995.

[2] M. Pedram, "Power Minimization in IC Design: Principles and Applications," *ACM Trans. on Design Automation of Electronic Systems*, vol. 1, no. 1, pp. 3-56, Jan. 1996.

[3] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Trans. on VLSI Systems*, vol. 2, no. 4, pp.446-455, December 1994.

[4] F. N. Najm, "Power Estimation Techniques for Integrated Circuits," *Proc. of ICCAD*, pp. 492-499, 1995.

[5] F. N. Najm, "Towards a High-Level Power estimation Capacility," *Proc. of ISLPD'95*, pp. 87-92, 1995.

[6] D. Marculescu, R. Marculescu, and M. Pedram, "Information Theoretic Measures of Energy Consumption at Register Transfer Level," *Proc. of ISLPD'95*, pp. 81-96, 1995.

[7] E. Kellman, "A Formula for Logic Network Cost," *IEEE Trans. on Computers*, vol. C-17, pp. 881-884, September 1968.

[8] L. Hellerman, "A Measure of Computational Work," *IEEE Trans. on Computers*, vol. C-21, pp. 439-446, May 1972.

[9] R. W. Cook and M. J. Flynn, "Logical Network Cost and Entropy," *IEEE Trans. on Computers*, vol. C-22, pp. 823-826, September 1973.

[10] N. Pippenger, "Information Theory and the Complexity of Boolean Functions," *Math Syst. Theory*, vol. 10, pp. 129-167, 1977.

[11] K.-T. Cheng and V. Agrawal, "An Entropy Measure for the Complexity of Multi-Output Boolean Functions," in *Proc. of 27th Design Automation Conf.*, pp. 302-305, 1990.

[12] K. Mase, "Comments on A Measure of Computational Work and Logical Network Cost and Entropy," *IEEE Trans. on Computers*, vol. C-27, pp. 94-95, January 1978.

[13] R. K. Brayton and et al., "MIS: A Multiple Level Logic Optimization System," *IEEE Trans. on CAD*, vol. CAD-6, pp. 1062-1081, November 1987.

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