Towards a High-Level Power Estimation Capability †

Mahadevamurty Nemani and Farid N. Najm

ECE Dept. and Coordinated Science Lab.
University of Illinois at Urbana-Champaign
Urbana, Illinois 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. With such early warning, the designer can explore design trade-offs at a higher level of abstraction than previously possible, reducing design time and cost. 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 tested on a variety of circuits. The empirical results to be presented are very promising and demonstrate the feasibility and utility of this approach.

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

Submitted to the IEEE Transactions on CAD, 1995.

1. Introduction

The high device count and operating frequency of modern integrated circuits has led to unacceptably high levels of chip power consumption. Modern microprocessors have power consumption specifications that can easily exceed 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. While several approaches have been proposed for gate-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. This paper is an extended version of [18].

The remaining sections are organized as follows. In section 2 we give a brief review of the concept of entropy and its application to logic circuits. Our model for average activity in terms of input/output entropy is given in section 3, and is then verified in section 4 against empirical data. Section 5 provides a discussion of area estimation from entropy and presents improved bounds for area prediction. Finally, conclusions are presented in section 6.

2. 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:

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

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

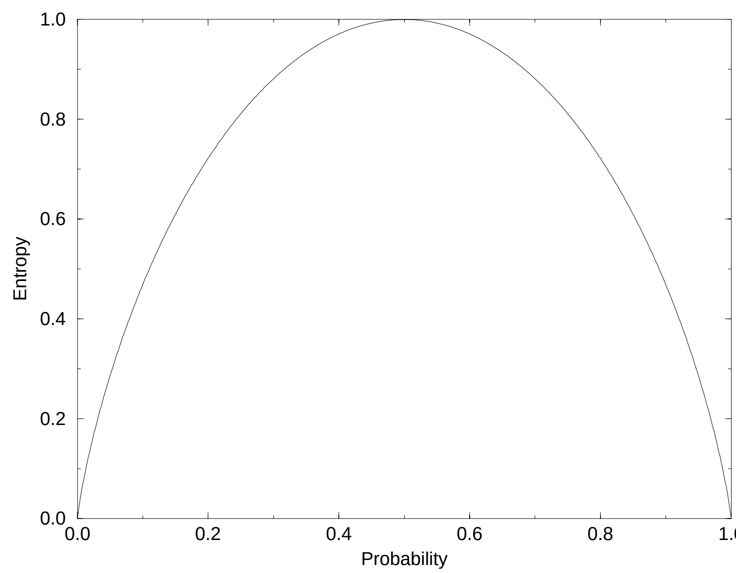

Figure 1. The entropy of a Boolean variable.

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:

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

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 $H(Y) \leq H(X)$, 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 gate-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 ($n \leq 10$), it was empirically observed [8] that $2^n H(Y)$ provides a good measure of area. We will show in section 5 that this model breaks down over a range of realistic input counts from 25 to 200. In that section, we will also give two entropy-based new bounds on the area that perform better than the above.

3. Power Estimation

We restrict ourselves to the common static fully-complementary 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 4. 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 $\sum_{i=1}^{N} C_i$.

It also seems inevitable that both $\mathcal{D}$ and $\mathcal{A}$ will be estimated only from knowledge of the input/output behavior. The main result of this paper will be a relationship between the average density $\mathcal{D}$ and entropy. We will also give (in section 5) two new bounds on the area that perform better than existing methods. 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 (part of step 2). Step 3 is easy, given the clock frequency and the results of steps 1 and 2.

3.1 Entropy from RTL Trace

The entropy $H(x_i)$ of a single input or output node x_i can be easily computed from the definition (1) once the node probability has been found. The probabilities of the primary inputs of the sequential machine (nodes u_i in Fig. 2) are assumed to be provided by the user, or can otherwise be easily extracted from an RTL trace by counting the proportion of 1s. The other inputs of the combinational block are latch outputs, whose probabilities can be

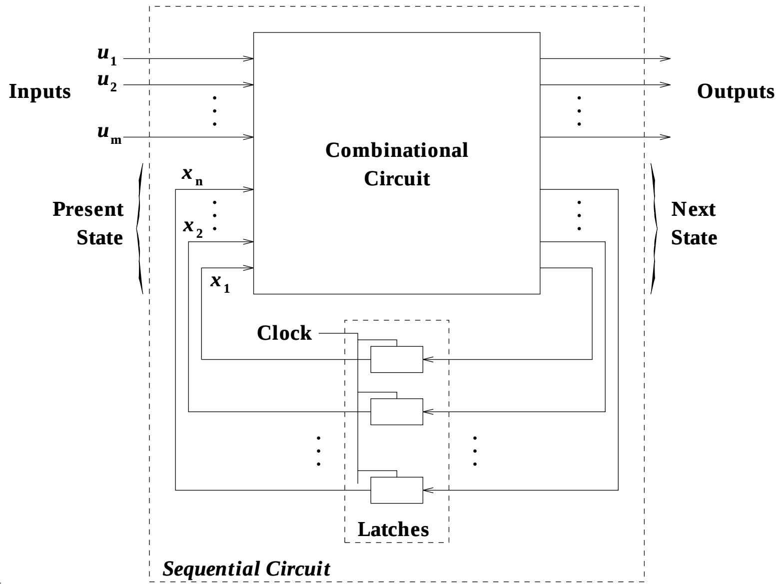

Figure 2. A general synchronous sequential circuit.

obtained as in [13, 14], or from an RTL trace using Monté Carlo analysis as proposed in [11]. The same analysis also yields the probabilities of the latch inputs and of the other outputs of the combinational block.

For area estimation, it has been found [8] that the entropy of the output Boolean *vector* plays a key role. 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 (3). Instead, one can efficiently estimate an upper bound on the entropy based on the relation:

$$H(X) \leq \sum_{i=1}^{n} H(x_i) \tag{10}$$

where equality occurs when the signals x_i are independent [9]. Thus, if the bits in a Boolean vector are not too correlated, one can make the approximation:

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

We refer to this as a *zeroth order* approximation and denote it by $H^0(X) = \sum_{i=1}^{n} H(x_i)$. Other higher-order approximations are possible, as will be shown in section 5.

3.2 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.

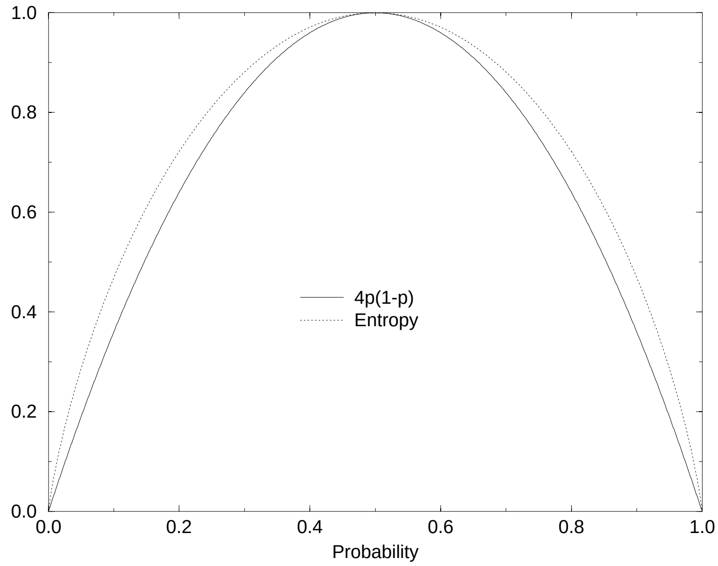

Figure 3. The relation between activity and entropy.

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):

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

In the following, we will see how the average internal node entropy of a combinational circuit can be computed from its input/output entropy. We will start the analysis by considering the variation of the internal entropy in a completely specified gate-level implementation of the circuit. From this, we will carefully abstract away (using a series of approximations) all aspects of circuit structure to end up with a model that depends only on the circuit input/output properties.

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*. Since level numbers are gate attributes, not node attributes, it will be helpful to define the notion of a circuit cross-section, as follows. A node which is the output of some gate g is said to be *generated* at the level of g. A primary input node is said to be generated at level 0. A node which is the input of some gate g is said to be *used* at the level of g. A primary output node is said to be used at level $K+1$. Thus, every circuit node is generated at some unique level and used at possibly several other levels. For every $i = 0, 1, 2, \ldots, K$, define the *set of nodes in cross-section* i, $\mathcal{S}_i$, as the set of all circuit nodes that are generated at levels less than or equal to i and used at levels greater than i.

3.2.1 Definition of $\hat{H}(i)$

Based on the notion of cross-section, we define $H(i)$ to be the sum of node entropies in the set $\mathcal{S}_i$, called the *cumulative entropy at cross-section* i or, simply, the *entropy at cross-section* i. Thus $H(K)$ is the sum of entropies of the primary output nodes (next state vector + output vector, in the case of a sequential circuit), denoted by H_o. Likewise, $H(0)$ is the sum of entropies of the primary input nodes (present state vector + primary inputs vector, in the case of a sequential circuit), denoted by H_i.

Let $W(i)$ be the number of nodes in cross-section i, i.e., the number of elements in $\mathcal{S}_i$. This we will call the circuit *width* at that cross-section. Thus $W(K)$ is the number of primary output nodes, which we denote by m (the output width). And $W(0)$ is the number of primary input nodes, denoted by n (the input width). Define $H_{avg}(i)$ as:

$$H_{avg}(i) = \frac{H(i)}{W(i)} \tag{13}$$

so that $H_{avg}(i)$ is the average entropy per node at cross-section i.

Given only a high-level specification, the width $W(i)$ at internal cross-sections of the circuit is unknown. Consider the linear width model:

$$\hat{W}(i) = m + (n - m)\left(1 - \frac{i}{K}\right) \tag{14}$$

We are not actually assuming that the circuit width is linear like this, but will only use the model (14) as a scaling factor as we look for a reasonable entropy model, as we shall see. Note that $\hat{W}(0) = n = W(0)$ and $\hat{W}(K) = m = W(K)$. Based on this, we now define the quantity $\hat{H}(i)$ as follows:

$$\hat{H}(i) = H_{avg}(i)\hat{W}(i) \tag{15}$$

so that $\hat{H}(i)$ is the entropy at cross-section i corresponding to a linear width model. Note that $\hat{H}(0) = H_i$ and $\hat{H}(K) = H_o$.

3.2.2 Quadratic model for $\hat{H}(i)$

We have empirically observed that $\hat{H}(i)$ varies *quadratically* with depth. This is shown in Figs. 4(a), 4(b), and 4(c) where we show, for some example circuits, the actual variation of $\hat{H}(i)$ with depth and compare it with the quadratic model for $\hat{H}(i)$, namely $H^q(i)$, which can be written in terms of H_i and H_o as follows:

$$H^q(i) = H_o + (H_i - H_o)\left(1 - \frac{i}{K}\right)^2 \tag{16}$$

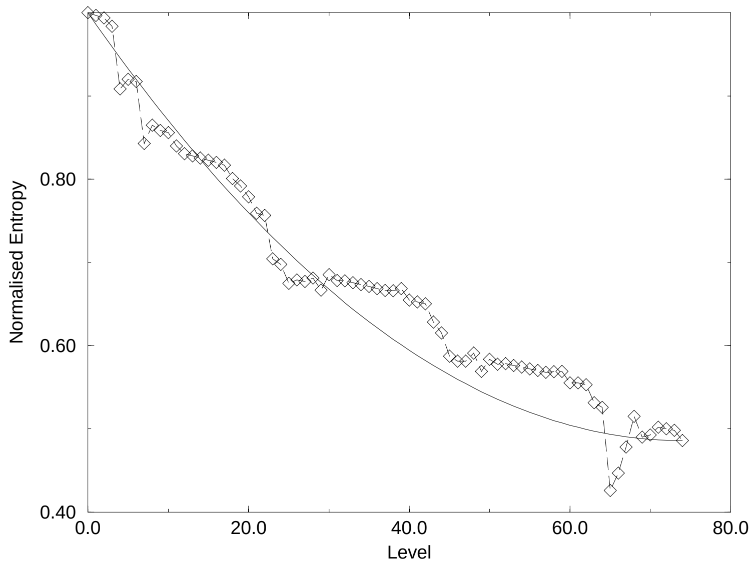

Figure 4(a). Comparison of H^q with $\hat{H}$ for S713 .

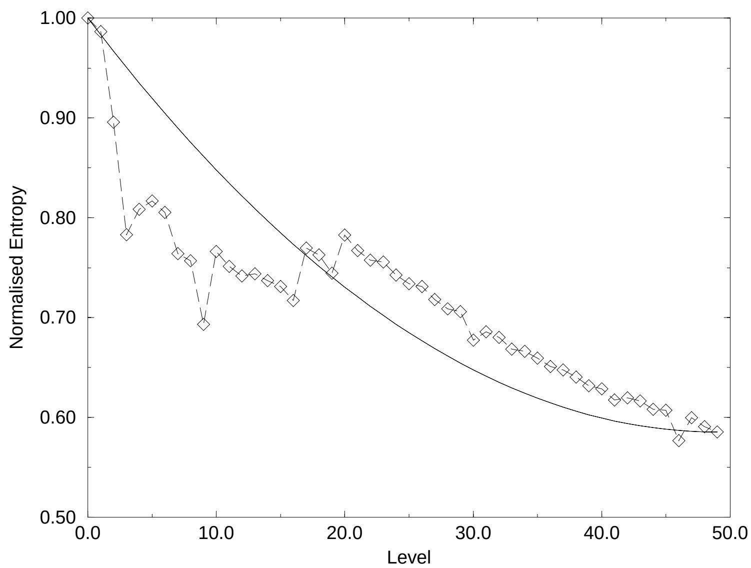

Figure 4(b). Comparison of H^q with $\hat{H}$ for C5315.

The plots show that there is a good agreement between the quadratic model, $H^q(i)$, and the actual variation of $\hat{H}(i)$. The deviation from the quadratic model observed at the input side is due to the fact that the actual cross-sectional entropy falls faster than the quadratic model in this region. The deviation in the middle is due to the fact that the actual cross-sectional entropy increases due to recombination or reconvergent fanout while the cross-sectional entropy as predicted by the quadratic model continues to decrease. Towards the

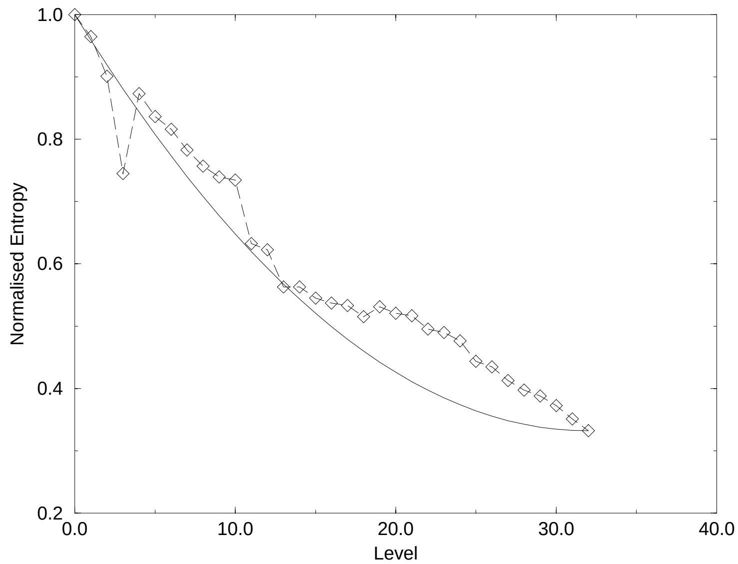

Figure 4(c). Comparison of H^q with $\hat{H}$ for C2670.

output side, $\hat{H}$ seems to fall linearly. This implies that the average entropy per node in a cross-section, $H_{avg}(i)$, is approximately constant towards the output end.

To further verify the quadratic model, the root mean square (RMS) error between the quadratic model and the actual variation was measured for all the ISCAS-85 and ISCAS-89 benchmark circuits for input probabilities ranging from 0.1 to 0.9. A histogram depicting the errors obtained is shown in Fig. 5. The histogram indicates that H^q is a good approximation of $\hat{H}$. In the cases where the RMS error was large it was observed that the quadratic model overbounds the actual data. This means that the quadratic approximation is conservative, and would err on the side of higher activity.

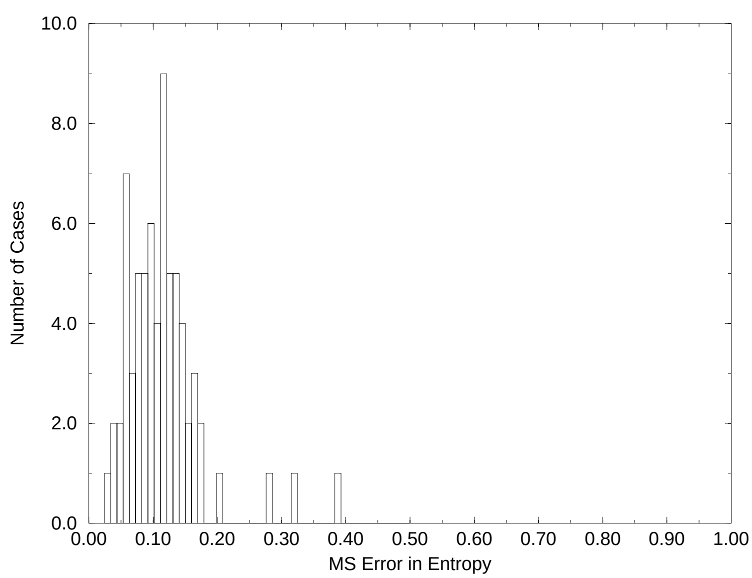

Figure 5. Root-Mean-Square Error between H^q and $\hat{H}$.

3.2.3 Average entropy model

Having empirically demonstrated the fact that $\hat{H}$ decreases *quadratically* with circuit depth, we now derive a model for computing the average node entropy from the input and output entropies as follows. By the definition of average entropy per node, $\mathcal{H}$, we have:

$$\mathcal{H} = \frac{1}{N} \sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j) \tag{17}$$

where N is the number of logic gates in the circuit, and $\mathcal{G}_i$ is the set of nodes x_j that are outputs of gates at level i. We define $\mathcal{G}_0$ to be equal to the empty set. Let N_s be defined as:

$$N_s = \sum_{i=0}^{K} W(i) \tag{18}$$

It then follows from the definitions of N_s and $H(i)$ that $N_s \geq N$ and $H(i) \geq \sum_{\mathcal{G}_i} H(x_j)$. Thus,

$$\frac{1}{N_s} \sum_{j=0}^{K} H(j) = \frac{\sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j) + \sum_{j=0}^{K} \delta H(j)}{N + \sum_{j=0}^{K} \delta N(j)} \tag{19}$$

where $\delta H(j)$ is the sum of entropies of those nodes in $\mathcal{S}_j$ but not in $\mathcal{G}_j$ and $\delta N(j)$ is the *number* of such nodes. Now let $\delta H = \sum_{j=0}^{K} \delta H(j)$ and $\delta N = \sum_{j=0}^{K} \delta N(j)$. If δN is small relative to N, we have the approximation:

$$\frac{1}{N_s} \sum_{j=0}^{K} H(j) \approx \frac{1}{N} \sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j) + \frac{1}{N} \left[\delta H \left(1 - \frac{\delta N}{N} \right) - \sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j) \frac{\delta N}{N} \right] \tag{20}$$

which is the result of using a Taylor series expansion of $1/(1 + \delta N/N)$ and dropping the high order terms. For large N, we can further approximate the above expression and write the equation as:

$$\frac{1}{N_s} \sum_{j=0}^{K} H(j) \approx \frac{1}{N} \sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j) \tag{21}$$

When the condition $\delta N \ll N$ does not hold, the Taylor approximation can not be used. In spite of this, (21) can be derived from (19) by rewriting (19) as follows:

$$\frac{1}{N_s} \sum_{j=0}^{K} H(j) = \frac{\sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j) \left[1 + \frac{\delta H}{\sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j)} \right]}{N \left[1 + \frac{\delta N}{N} \right]}$$

Now let $k_1 = 1 + \frac{\delta H}{\sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j)}$ and $k_2 = 1 + \frac{\delta N}{N}$. It must be observed that if

$$\frac{\delta H}{\delta N} \approx \frac{\sum_{i=1}^{K} \sum_{\mathcal{G}_i} H(x_j)}{N}$$

then $\frac{k_1}{k_2} \approx 1$. The correlation between k_1 and k_2 was measured for ISCAS-85 and ISCAS-89 circuits for input probabilities ranging from 0.1 to 0.9 and the results are shown in Fig. 6. It can be seen from this figure that $\frac{k_1}{k_2} \approx 1$ and hence, when δN is large, it is reasonable to assume that $\frac{k_1}{k_2} \approx 1$. It then follows that for large δN (19) can be approximated by (21).

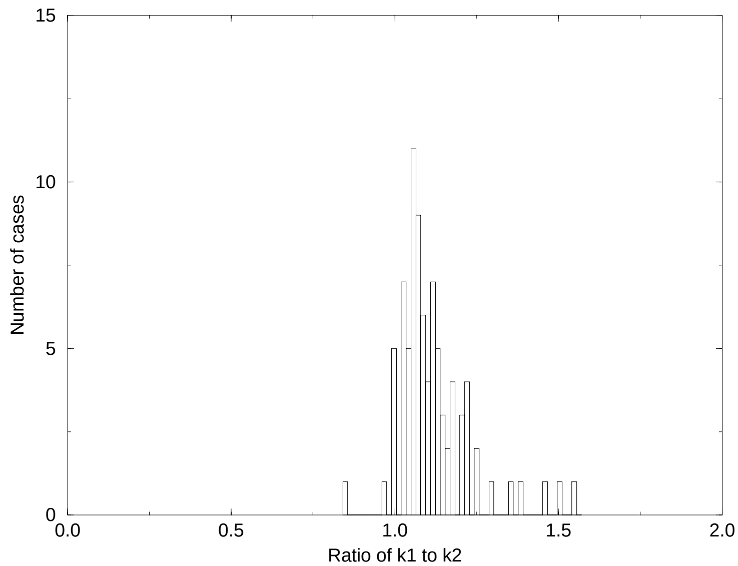

Figure 6. Histogram of the ratio $\frac{k_1}{k_2}$.

The empirical results presented in the next section indicate that the above assumptions are quite reasonable for large circuits. From (17) and (21), it thus follows that:

$$\mathcal{H} \approx \frac{1}{N_s} \sum_{j=0}^{K} H(j) \tag{22}$$

The right hand side of the above equation can be further approximated as follows:

$$\frac{1}{N_s} \sum_{j=0}^{K} H(j) = \frac{\sum_{j=0}^{K} H_{avg}(j) W(j)}{\sum_{j=0}^{K} W(j)} \approx \frac{\sum_{j=0}^{K} H_{avg}(j) \hat{W}(j)}{\sum_{j=0}^{K} \hat{W}(j)} = \frac{1}{N_l} \sum_{j=0}^{K} \hat{H}(j) \tag{23}$$

where N_l is the total number of nodes corresponding to a linear width model, defined as:

$$N_l = \sum_{j=0}^{K} \hat{W}(j) \tag{24}$$

The errors between $\frac{1}{N_s} \sum_{j=0}^{K} H(j)$ and $\frac{1}{N_l} \sum_{j=0}^{K} \hat{H}(j)$ were measured for the ISCAS-85 and ISCAS-89 circuits for input probabilities ranging from 0.1 to 0.9. The results obtained are summarized in the histogram shown in Fig. 7.

It can be seen from the histogram that the error due to the above approximation is quite small. This implies that the above approximation is quite good. Thus, we have reduced the

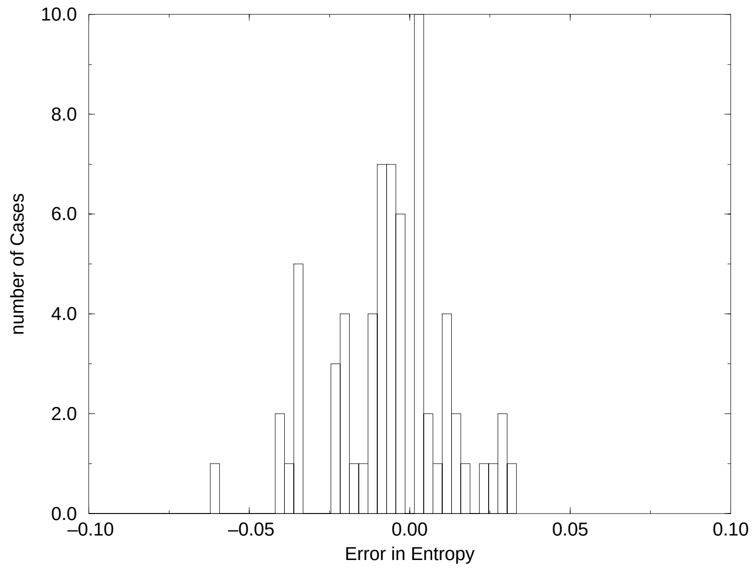

Figure 7. Error between $\frac{1}{N_s}\sum_{j=0}^{K} H(j)$ and $\frac{1}{N_l}\sum_{j=0}^{K} \hat{H}(j)$.

problem of computing $\mathcal{H}$ for the actual circuit to computing the average of $\hat{H}(j)$ for the linear width model $\hat{W}(j)$. But we have already demonstrated that $\hat{H}(j)$ can be approximated by the quadratic model $H^q(j)$. This implies that:

$$\frac{1}{N_l}\sum_{j=0}^{K} \hat{H}(j) \approx \frac{1}{N_l}\sum_{j=0}^{K} H^q(j) \tag{25}$$

Thus it follows from (22), (23), and (25), that:

$$\mathcal{H} \approx \frac{1}{N_l}\sum_{j=0}^{K} H^q(j) \tag{26}$$

Now, if we denote the average width of the linear model by $\hat{\mathcal{W}}$, so that:

$$\hat{\mathcal{W}} = \frac{1}{K+1}\sum_{j=0}^{K} \hat{W}(j) = (n+m)/2 \tag{27}$$

then, from (24), (26), and (27), it follows that:

$$\begin{aligned} \mathcal{H}\hat{\mathcal{W}}(K+1) &\approx \sum_{j=0}^{K} H^q(j) = (K+1)H_o + (H_i - H_o)\sum_{j=0}^{K}\left(1 - \frac{j}{K}\right)^2 \\ &= (K+1)H_o + (H_i - H_o)\frac{1}{K^2}\sum_{j=0}^{K}(K-j)^2 = (K+1)H_o + (H_i - H_o)\frac{1}{K^2}\sum_{k=1}^{K} k^2 \qquad (28) \\ &= (K+1)H_o + (H_i - H_o)\frac{K(K+1)(2K+1)}{6K^2} \end{aligned}$$

From which, it follows that:

$$\mathcal{H}\hat{\mathcal{W}} = \frac{4K-1}{6K}H_o + \frac{2K+1}{6K}H_i \approx \frac{2}{3}H_o + \frac{1}{3}H_i \tag{29}$$

where the last approximation is based on the fact that 1 is negligible compared to $2K$ and $4K$, given that the circuit depth K can be large for large circuits. This leads to:

$$\mathcal{H}\hat{\mathcal{W}} \approx \frac{H_i + 2H_o}{3} \tag{30}$$

which does not depend on circuit depth (a must, so as to be applicable to a high level representation). Using (27), we finally arrive at our main result:

$$\mathcal{H} \approx \frac{2/3}{n+m}\left(H_i + 2H_o\right) \tag{31}$$

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 $\mathcal{H}$, (31), works quite well for a broad range of circuits. The empirical results presented in the next section will be based on this expression.

4. 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 (31). To use this technique, we estimate H_i and H_o from their definitions and then use (31). We tested the technique on isolated combinational circuit blocks whose input probabilities are user-specified. Normally, these 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 [15].

In order to assess the accuracy of the technique, we need an accurate measure of average node activity, obtained from a gate-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]

according to $D(x) = 2p(1-p)/T_c$, 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 (31), 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 (31) against the zero-delay simulation results.

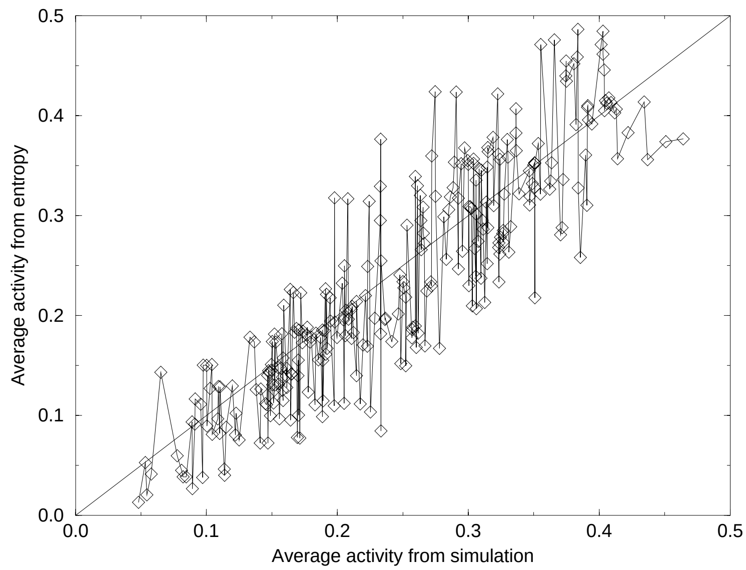

Figure 8. Comparison with zero-delay simulation results.

The results of testing against the zero-delay analysis results are shown in Fig. 8 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 (31), requires only 14 cpu seconds for a 20,000 gate circuit (on a SUN sparc-10).

The effect of capacitance is not included in the data shown Fig. 8 (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. 9. The results shown are for the same circuits used in Fig. 8, but only for an input probability value of 0.5. The results show slightly more spread than Fig. 8, due to the effect of the node capacitance distribution.

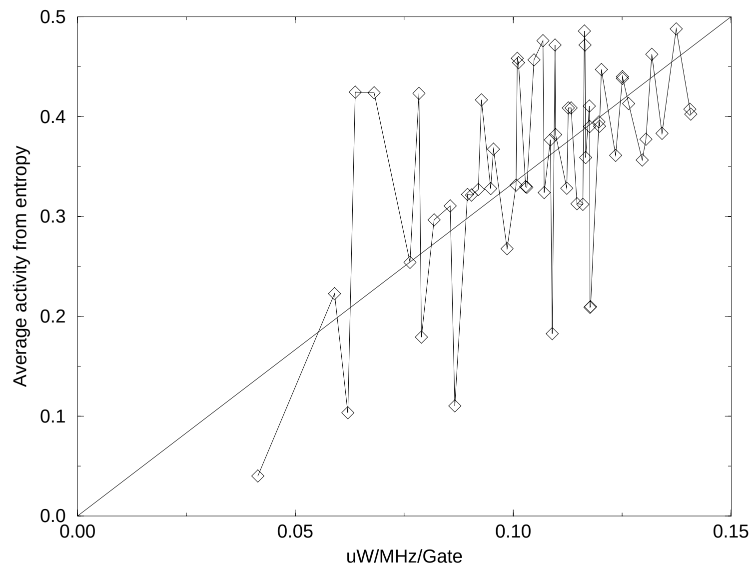

Figure 9. Comparison with zero-delay power/area simulation results.

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. 10. 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. 11.

5. Area Measurement

In the previous sections we presented an approach for measuring the average activity of a circuit using entropy. In order to estimate the average power consumed by a circuit we also need an estimate of the total area (capacitance) consumed by a circuit (see equation (7)). Thus, one has to come up with techniques to measure the area consumed by a circuit from its high level description.

The importance of entropy in the computation of area complexity of a Boolean function has been known for some time. However most of the work focuses on characterization of the

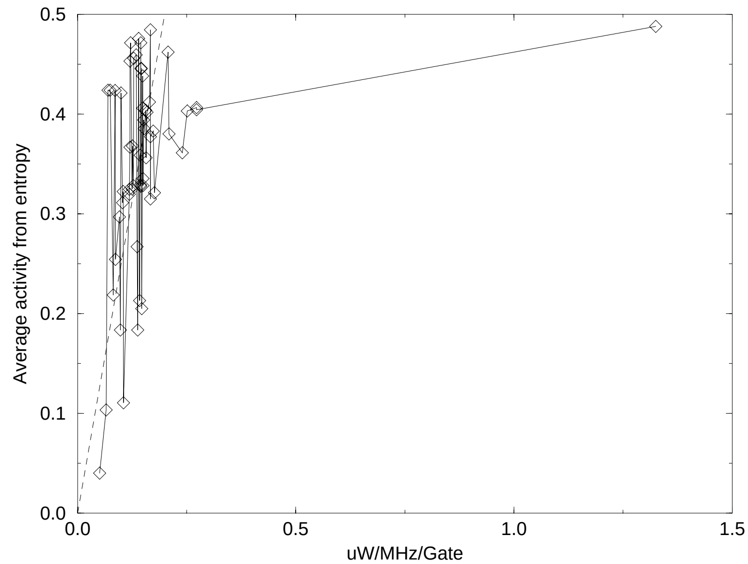

Figure 10. Comparison with general-delay power/area simulation results.

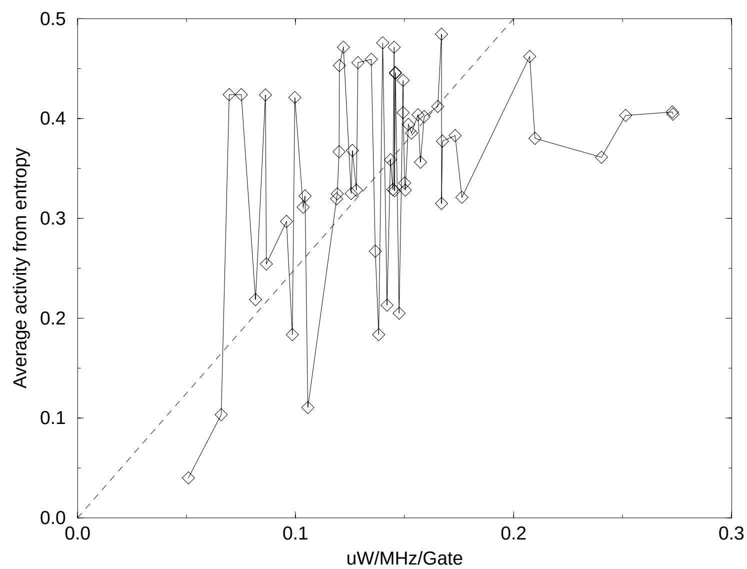

Figure 11. Comparison with general-delay power/area simulation results.

area consumed by single output functions. In [8], Cheng et. al proposed a model for the area complexity of a multi-output Boolean function given by the following expression

$$\mathcal{A} = (1-d)k2^n H \tag{32}$$

Here, $\mathcal{A}$ is a measure of the area complexity of the circuit in terms of the literal count, d is the fraction of don't cares in the function specification, n is the number of inputs, H is the entropy of the output vector and k is some proportionality constant. In this paper we have used gate count as a measure of area consumed by a circuit. In [17], Muller showed that the type of gates used affects the area complexity by a scaling constant, asymptotically. We assume that this fact holds true for all n.

According to the model of Cheng et. al [8], the area complexity increases exponentially in the number of inputs. This implies that circuits with input counts of the order of a few hundreds, which is quite common, would have an exponentially large area requirement. But this does not seem to be the case as can be seen from Table 1, where the constant k is shown to vary widely as the number of inputs changes. The variation in k is so large that the model (32) is neither a good area estimator, nor a good area upper bound.

TABLE 1

EVALUATION OF PROPORTIONALITY CONSTANT IN AREA MODEL OF CHENG et. al

CIRCUIT	INPUTS	#GATES	k
s400	25	164	2.92e-7
s713	54	393	1.12e-15
c2670	157	1193	1.78e-46
c5315	178	2307	7.85e-53

The fact that circuits with large numbers of inputs and outputs are typically designed to have reasonable (at least, not exponentially large) area suggests that the model (32) is not applicable to practical circuits, and needs to be improved.

As a first step towards improving the area model (32), we have developed an efficient scheme for estimating a more accurate estimate of the entropy of a Boolean vector. This estimate takes the form of an upper bound on the entropy that is is more accurate than the zeroth order approximation (11), because it does not completely ignore the bit correlations. Based on this, we then verified that two entropy-based simple bounds on the area seem to work well in practice well into the range of input counts where (32) breaks down. Both these topics are discussed below.

5.1 Entropy Bound Computation

To compute a more accurate upper bound on the vector entropy, we will take into account pair-wise correlations between the bits. Before getting into the details of the computation, a definition of conditional entropy is in order. Let $Y_1 \in \mathcal{Y}_1$ and $Y_2 \in \mathcal{Y}_2$ be scalar random variables with joint probability density function given by $p(y_1, y_2)$ and a conditional density function given by $p(y_1 \mid y_2)$. Then the conditional entropy [9] of Y_1 given Y_2, denoted by $H(Y_1 \mid Y_2)$, is defined as:

$$H(Y_1 \mid Y_2) = -\sum_{y_2 \in \mathcal{Y}_2} \sum_{y_1 \in \mathcal{Y}_1} p(y_1, y_2) \log_2 p(y_1 \mid y_2) \qquad (33)$$

Intuitively, conditional entropy measures the additional information required to encode Y_1 having specified (encoded) Y_2.

Now we discuss the calculation of the upper bound on $H(Y)$, where $Y = [y_1, y_2, \ldots, y_n]$ is a Boolean output vector. It can be shown (refer to [9]) that:

$$H(Y) = H(y_1) + H(y_2 \mid y_1) + H(y_3 \mid y_2, y_1) + \cdots + H(y_n \mid y_{n-1}, \ldots, y_1) \tag{34}$$

where the above equation is true for all orderings of the bits of vector Y. Also:

$$H(y_k \mid y_m, y_{m-1}, \ldots, y_1) \leq H(y_k \mid y_j), \qquad j = 1, 2, \ldots, m \tag{35}$$

Using the above facts one can write an upper bound on $H(Y)$ as:

$$H(Y) \leq H(y_1) + H(y_2 \mid y_1) + H(y_3 \mid y_2) + \cdots + H(y_n \mid y_{n-1}) \tag{36}$$

Since the expansion of $H(Y)$ in terms of conditional entropies is independent of the ordering of y_i in Y, we get that the above inequality must hold for all orderings of the components of vector Y. Let $\mathcal{Z}$ be the set of all possible orderings of y_i, $i = 1, 2, \ldots, n$ in Y. Further let $H^1_{opt}(Y)$ be defined as:

$$H^1_{opt}(Y) = \min_{\mathcal{Z}} H(Y) \tag{37}$$

Then it follows that:

$$H^1_{opt}(Y) \leq H(y_1) + H(y_2 \mid y_1) + H(y_3 \mid y_2) + \cdots + H(y_n \mid y_{n-1}) \tag{38}$$

Also,

$$H(Y) \leq H^1_{opt}(Y) \tag{39}$$

Note that if $y_i, i = 1, 2, \ldots, n$ are not independent, then the above bound is strictly smaller than the sum of the individual entropies, i.e.,

$$H(Y) \leq H^1_{opt}(Y) < H^0(Y) \tag{40}$$

where:

$$H^0(Y) = \sum_{i=1}^{n} H(y_i). \tag{41}$$

Thus $H^1_{opt}(Y)$ is called the optimal first order bound on entropy. Since it is computationally expensive to find the optimal ordering, we will resort to a heuristic that orders the variables based on their *correlation coefficients*. Given this ordering, we will refer to the resulting entropy upper bound (36) as a first order bound, denoted $H^1(Y)$.

The following briefly describes how the heuristic works. First, we compute all the correlation coefficients $\rho(y_i, y_j), i \neq j$, and set $\rho(y_i, y_i) = 0$, as we are not interested in these. To compute these coefficients, it suffices to find the joint probability of every two bits, i.e., the probability that $y_i y_j = 1$. These probabilities can be found at the same time that signal probabilities are computed, using Monté Carlo analysis [11]. We then pick the element of the resulting correlation matrix that has the largest absolute value. The row and column indices of this element provide us the first two elements in our ordering, i.e., y_1^o and y_2^o. We then set the correlation coefficients $\rho(y_i, y_1^o) = 0$, $i = 1, 2, \ldots, n$ as we do not want to pick these pairs. Now we pick out the element that has the largest correlation coefficient (in absolute value) with y_2^o. This is the third element in our ordering. Then the correlation coefficients $\rho(y_i, y_2^o)$, $i = 1, 2, \ldots, n$ are set to zero. This procedure is repeated until all elements are exhausted. The algorithm has $\mathcal{O}(n^2)$ time and space complexity.

In Table 2, we perform a comparative study between the first order bound on entropy, $H^1(Y)$, computed using the above heuristic, and the zeroth order bound, $H^0(Y)$, on a few circuits from ISCAS-85 and ISCAS-89 benchmarks. It can be seen that, on the average, $H^1(Y)$ offers a 16% improvement over $H^0(Y)$. In the next section, we use this bound to obtain empirical upper and lower bounds on the area complexity of a combinational circuit.

TABLE 2

COMPARISON OF FIRST AND ZERO ORDER BOUNDS ON ENTROPY

(ISCAS-85 and ISCAS-89 CIRCUITS)

CIRCUIT Name	INPUTS	OUTPUTS	ENTROPY FIRST	ENTROPY ZERO	% IMPROV
c880	60	26	16.47	17.48	5.76
c1908	33	25	24.24	24.63	1.93
c2670	157	64	36.75	52.34	29.77
c3540	50	22	17.92	19.42	7.70
c5315	178	123	76.64	104.05	26.34
c6288	32	32	31.03	31.17	0.43
c7552	208	107	83.01	99.47	16.54
s382	24	27	15.69	19.97	21.39
s510	25	13	9.02	10.32	12.70
s526	24	27	20.52	22.84	10.16
s713	54	42	19.65	26.23	25.08
s953	22	29	6.84	10.05	31.94
s1196	31	31	15.83	18.23	13.20
s5378	214	156	76.06	95.10	20.00
s9234.1	247	250	186.72	225.91	17.35
Average					16.02

5.2 Area Bounds

It is well known that (see [5]) as the number of inputs tends to infinity, the area complexity of a circuit grows exponentially. However, as our data has shown, this asymptotic bound is quite loose for practical circuits. Thus one has to come up with better models to predict the area of a circuit. Here, we present a lower bound and an upper bound, which were arrived at empirically, that seem to describe the area behaviour of the ISCAS-85 and ISCAS-89 benchmark circuits, much better than (32).

We begin by assuming that the area of a circuit depends only on the output entropy (when the inputs are assumed to be independent and equi-probable with probability 0.5) and the number of inputs in a circuit in the following way:

$$\mathcal{A} \propto H(Y)f(n) \tag{42}$$

where $H(Y)$ is the entropy of the output vector and $f(n)$ is some function of the number of inputs n. This model is along the same lines of previous area models (see [2,4,5,8]). In our work, we have characterized the function $f(n)$ empirically, to obtain functions $f_{low}(n)$ and $f_{upp}(n)$ such that:

$$k_1 H(Y) f_{low}(n) \leq \mathcal{A} \leq k_2 H(Y) f_{upp}(n) \tag{43}$$

We have found that the following bounds work well:

$$0.4 H^1(Y)(\frac{n}{\log_{10} n}) \leq \mathcal{A} \leq 2H^1(Y)(n \log_{10} n) \tag{44}$$

where we have used $H^1(Y)$ instead of $H(Y)$, because $H(Y)$ is too expensive to compute. These bounds suggest that the circuit area grows sublinearly for medium sized circuits. These bounds are compared with the actual area (gate count) consumed by the circuits in Table 3. Of course, these bounds are not as close to each other as one would like, and we do not suggest that they be used *directly* for area estimation. Nevertheless, these bounds are valuable because they are valid over a wide range of circuits and input node counts. In this, they represent a significant improvement over the previous model (32) which completely breaks down when the inputs count is around 200.

6. 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. We also presented an algorithm to estimate the entropy of a vector which was used in obtaining improved bounds on the area complexity of combinational circuits.

References

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

TABLE 3

COMPARISON OF LOWER AND UPPER BOUNDS ON AREA WITH ACTUAL AREA REQUIREMENTS

(ISCAS-85 and ISCAS-89 CIRCUITS)

CIRCUIT Name	INPUTS #	AREA #GATES	LOWER BOUND	UPPER BOUND
s208.1	18	104	65	405
s298	17	119	85	515
s953	22	395	57	409
s820	23	289	36	267
s444	24	181	111	841
s382	24	158	108	823
s344	24	160	107	876
s526	24	193	141	1075
s400	25	164	120	935
s510	25	211	81	631
s1196	31	529	132	1169
c6288	32	2416	265	2394
c1908	33	880	211	2429
s420.1	34	218	151	1412
c432	36	160	53	509
c499	41	202	326	3386
c1355	41	546	325	3383
c3540	50	1669	211	2432
s713	54	393	244	2919
c880	60	383	222	2800
s838.1	66	446	478	6328
c2670	157	1193	1049	20228
c5315	178	2307	2429	49191
c7552	208	3512	2948	63133
s5378	214	2779	2785	60492
s9234.1	247	5844	7705	176458

[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, Jan. 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*, pp. 302–305, 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*, pp. 12–17, 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*, pp. 18–23, 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*, pp. 728–733, 1994.

[17] D. E. Muller, “Complexity in electronic switching circuits,” *IRE Transactions on Electronic Computers*, vol. 5, pp. 15–19, March 1956.

[18] F. Najm, “Towards a High-Level Power Estimation Capability,” *ACM/IEEE International Symposium on Low-Power Design*, pp. 87–92, 1995.