

RT-Level Power Analysis Using
Information Theoretic Measures

Diana Marculescu, Radu Marculescu
and Massoud Pedram

CENG Technical Report 95-25

Department of Electrical Engineering - Systems
University of Southern
Los Angeles, California 90089-2562
(213)740-4458

December 1995

RT-Level Power Analysis Using Information Theoretic Measures

Diana Marculescu, Radu Marculescu and Massoud Pedram

Department of Electrical Engineering - Systems

University of Southern California

Los Angeles, CA90089-2562

Phone: (213)-740-4437

e-mail: draiciu@usc.edu

December 1995

RT-Level Power Analysis Using Information Theoretic Measures

Abstract

The problem of estimating the power consumption at logic and register transfer levels is addressed from an information theoretical point of view. It is shown that the average switching activity can be predicted without simulation using either entropy or informational energy averages. Consequently, two new measures relying on these concepts are developed. The accuracy of these models is investigated using common benchmarks and results are reported.

I. INTRODUCTION

Modern design tools have changed the entire design process of digital systems. As a result, most of the systems are conceived and designed today at the behavioral/logic level, with little or no knowledge of the final gate level implementation and the layout style. In particular, designers are becoming more and more interested in register-transfer level (RTL) modules (adders, multipliers, registers, multiplexers) and strategies to put them together in order to build complex digital systems.

Power minimization in digital systems is not an exception to this trend. Having as soon as possible in the design cycle an estimate of power consumption, can save significant redesign efforts or even completely change the entire design architecture. Circuit and gate level power estimation techniques have been extensively explored [1][2]; they roughly fall in two categories: *simulative* [3]-[8], and *probabilistic* [9]-[17]. Generally speaking, simulation-based techniques can provide very accurate results, but the price we have to pay is too high; one can extract switching activity information by exhaustive simulation on small circuits, but it is unrealistic to rely on simulation results for larger circuits. On the other side, probabilistic techniques can provide sufficient accuracy with a reasonable amount of computational work. Despite the huge amount of efforts directed in both directions, the performance gap between simulative and probabilistic approaches has remained basically the same during the last few years.

Higher levels of abstraction have been also considered, but here many problems are still pending a satisfactory solution. At this level, consistency is more important than accuracy, that is, relative (as opposed to absolute) evaluation of different designs is often sufficient. Most of the high level prediction tools combine deterministic analysis with profiling and simulation in order to address data dependencies. Important statistics include the number of instructions of a given type, the number of bus, register and memory accesses and the number of I/O operations executed within a given period [18][19]. Analytic modeling efforts have been described in [20] where a parametric power model was developed for macromodules. However, the trade-off between flexibility and accuracy is still a challenging task as major difficulties persist due to the lack of precise information and the conceptual complexity which characterizes these levels.

Power dissipation in CMOS circuits comes from three sources: leakage currents which include the reverse-biased junction and subthreshold currents, short-circuit currents which flow due to the DC path between the supply rails during output transitions and capacitive switching currents which are responsible for charging and discharging of capacitive loads during logic transitions. In well-designed circuits with relatively high threshold voltages, the first two sources are very small compared to the last one. Therefore, to estimate the total power consumption of a module (in a gate level implementation), we may only account for the capacitive switching currents, yet achieve sufficient levels of accuracy [21]:

$$P_{avg} = \frac{f_{clk}}{2} \cdot V_{DD}^2 \cdot \sum_n (C_n \cdot sw_n) \quad (1)$$

where f_{clk} is the clock frequency, V_{DD} is the supply voltage, C_n and sw_n are the capacitive load and the average switching activity of gate n , respectively (the summation is performed over all gates in the circuit). As we can see, in this formulation the average switching activity per node is a key factor, and therefore its correct calculation is essential for accurate power estimation. Note that for the same implementation of a module, different input sequences may give rise to different switching activities at the circuit inputs and at the outputs of internal gates and consequently, completely different power values.

The problem of power estimation at the RT-level is different from that at the logic level: whilst at gate level it is desirable to determine the switching activity at each node (gate) in the circuit (Fig.1(a)), for RT-level designs an average estimate per module is satisfactory (Fig.1(b)). In other words, some accuracy may be sacrificed in order to obtain an acceptable power estimate early in the design cycle and at a significantly lower computational cost.

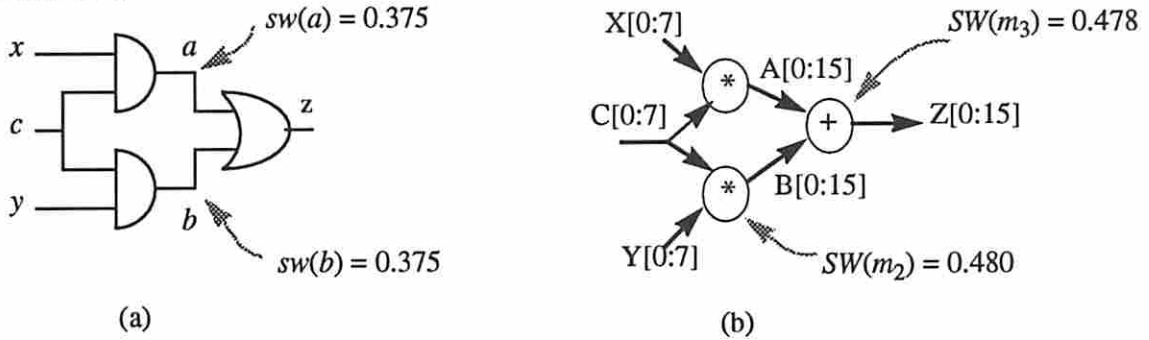


Fig.1: Power estimation issues at the logic and RT-levels

In the data-flow graph considered in Fig.1(b), the total power consumption may be estimated as: $P_{total} = P_{wires} + P_{modules}$. Usually, the interconnect power consumption is either estimated separately or included in the power consumption of the modules, therefore we can write:

$$P_{total} \approx \sum_{m_j \in M} P_{m_j} \propto \sum_{m_j \in M} (C_{m_j} \cdot SW_{m_j})$$
 where the summation is performed over the set of modules M used in the data-flow graph, and C_{m_j} , SW_{m_j} stand for the capacitance loading and the average switching activity of module m_j , respectively. Basically, what we propose is to characterize the average switching activity of a module (SW_{m_j}) through the average switching activity for a *typical signal line* in that module (sw_{avg}). More formally, for a generic module m_j having n internal lines (each characterized by its capacitance and switching activity values c_i and sw_i , respectively), we have:

$$P_{m_j} \propto \sum_{i=1}^n (c_i \cdot sw_i) \approx sw_{avg} \cdot \sum_{i=1}^n c_i = sw_{avg} \cdot C_{m_j} \quad (2)$$

We assume that module capacitances C_{m_j} are either estimated or taken from a library, therefore we concentrate on estimating the average switching activity per module. This is a quite different strategy compared to the previous work. The only other proposed method for power estimation at RT-level is simulative in nature, requires pre-characterization of the modules and may be summarized as follows: first, RT-level simulation is performed to obtain the average switching activity at the inputs of the modules and then, this switching activity is used to “modulate” a switched capacitance value (product of the switching activity and the physical capacitance) which is pre-computed and stored for each module in the library to obtain the power dissipation estimate [22]. Compared to this methodology, the distinctive feature of the present approach, is that *it does not require simulation*; its predictions are based only on the characteristics of the input sequence and some knowledge about the function and/or structure of the circuit (see Section IV for details).

In this paper, we address the problem of power estimation at the RT-level from an information theoretical point of view [23]. Traditionally, entropy has been considered a useful measure for solving problems of area estimation [24][25], timing analysis [26] and testing [27][28]. We propose two new measures for estimating the power consumption of each module based on *entropy* and *informational energy*. Our entropy/informational energy-based measures simply provide an approximation for the functional activity in the circuit without having to necessarily simulate the circuit. With some further simplifications, simple closed form expressions are derived and their value in practical applications is explored.

Note: We point out that, although this paper targets RT-level and behavioral design, it also presents as a by-product, a technique applicable to logic level designs. This is a first step in building a unified framework for power analysis from gate level to behavioral level.

The paper is organized as follows. Sections II and III introduce the main concepts and the motivation behind our model. In Section IV we present some practical considerations and finally, in Section V, we give the results obtained by analyzing a common data-path. We conclude by summarizing our main ideas and indicating possible extensions of the present work.

II. THEORETICAL FRAMEWORK

A. An Entropy-Based Approach

Let A_1, A_2, \dots, A_n be a complete set of events which may occur with the probabilities p_1, p_2, \dots, p_n that is:

$$p_k \geq 0, (\forall k), k = 1, 2, \dots, n \quad \sum_{k=1}^n p_k = 1 \quad (3)$$

We introduce the following definition:

Definition 1: (Self-Information)

The *self-information* of the event A_k , written $I(A_k)$, is defined as

$$I(A_k) = -\log_2(p_k) \quad (4)$$

where \log stands for \log_2 . The function $I(A_k)$ vs. $p_k = p(A_k)$ is plotted in Fig.2.

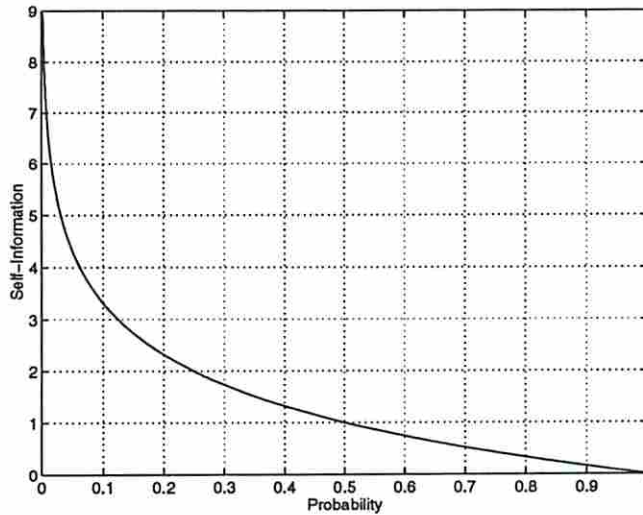


Fig.2: Self-information $I(A_k)$ vs. probability p_k .

Self-information can be interpreted as the amount of information provided by occurrence of the event A_k . According to this interpretation, the less probable an event is, the more information we receive when it occurs. A certain event (one that occurs with probability 1) provides no information at all, whereas an unlikely event provides a large amount of information. The measurement unit for I is the *bit* (*binary digit*); when $p_k = 1/2$, $I(A_k) = 1$ bit, so that 1 bit of information occurs on the choice of one from two equally likely events.

Let us consider an experiment where the outcome is unknown in the beginning; such an experiment exposes a probability finite field \mathcal{A}_n , completely characterized by the discrete probability distribution p_1, p_2, \dots, p_n . In order to quantify the content of information revealed by the outcome of such an experiment, Shannon introduced the concept of entropy [29].

Definition 2: (Entropy)

Entropy of a finite field \mathcal{A}_n (denoted by $H(\mathcal{A}_n)$) is given by:

$$H(\mathcal{A}_n) = H(p_1, p_2, \dots, p_n) = - \sum_{k=1}^n p_k \log p_k \quad (5)$$

In other words, entropy is the weighted average of self-information over all events of the field. We plot in Fig.3 the entropy of a Boolean variable as a function of its signal probability, that is $H(\mathcal{A}_2) = -p \cdot \log p - (1-p) \cdot \log(1-p)$.

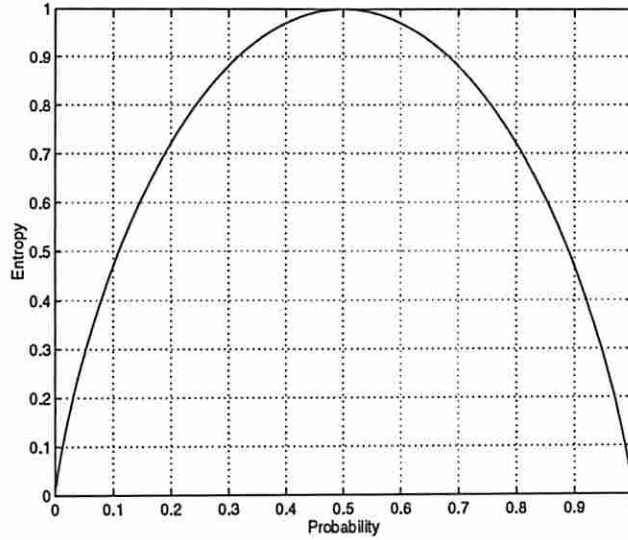


Fig.3: Entropy $H(\mathcal{A}_2)$ vs. probability

We pointed out earlier that self-information of an event increases as its uncertainty grows. Thus, entropy may be also regarded as a *measure of uncertainty* in the sense that the larger the entropy, the less certain the outcome of a random experiment in \mathcal{A}_n . Entropy is also a measure of the *information* contained in each event A_i (given by $I(A_i) = -\log(p_i)$).

Note: Information and uncertainty have the same quantitative measure (i.e. entropy), but different meanings. As information increases, the uncertainty decreases and vice versa. Indeed, uncertainty is equal to the amount of information which is needed to make the outcome of an experiment known.

First, we should note that $\log(p_k) \leq 0$ since $0 < p_k \leq 1$ and so $H(\mathcal{A}_n) \geq 0$. Thus, the entropy can never be negative. Second, let $p_1 = 1, p_2 = \dots = p_n = 0$. By convention, $p_k \log(p_k) = 0$ when $p_k = 0$ and hence, in this case, $H(\mathcal{A}_n) = 0$. Conversely, $H(\mathcal{A}_n) = 0$ implies that $p_k \log(p_k) = 0$ for all k , so that p_k is either 0 or 1. But only one p_k can be unity since their sum must be 1. Hence, entropy is zero if and only if there is complete certainty.

Definition 3: (Conditional Entropy)

Conditional entropy of some finite field \mathcal{A}_n with probabilities $\{p_i\}_{1 \leq i \leq n}$ with respect to \mathcal{B}_m (with probabilities $\{q_i\}_{1 \leq i \leq m}$) is defined as:

$$H(\mathcal{A}_n | \mathcal{B}_m) = - \sum_{j=1}^n \sum_{k=1}^m p_{jk} \cdot \log (p_{jk}/q_k) \quad (6)$$

where p_{jk} is the joint probability of events A_j and B_k . In other words, conditional entropy refers to the uncertainty left about \mathcal{A}_n when \mathcal{B}_m is known.

Definition 4: (Joint Entropy)

Given two finite fields \mathcal{A}_n and \mathcal{B}_m their joint entropy is defined as:

$$H(\mathcal{A}_n \times \mathcal{B}_m) = - \sum_{j=1}^n \sum_{k=1}^m p_{jk} \cdot \log (p_{jk}) \quad (7)$$

Based on these two concepts, one can find the information *shared* by two complete sets of events:

$$I(\mathcal{A}_n; \mathcal{B}_m) = H(\mathcal{A}_n) + H(\mathcal{B}_m) - H(\mathcal{A}_n \times \mathcal{B}_m)$$

which is called *mutual information* (or *transinformation*). Moreover, by using the above definitions, one can show that:

$$I(\mathcal{A}_n; \mathcal{B}_m) = H(\mathcal{A}_n) - H(\mathcal{A}_n | \mathcal{B}_m) \quad (8)$$

The Venn diagram for these relations is shown in Fig.4:

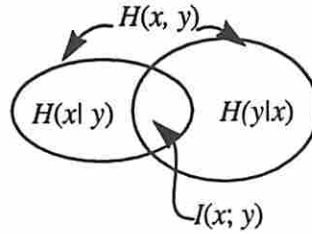


Fig.4: A set-theoretic representation of different entropies

Entropy satisfies the following basic properties [29]:

Property 1: Entropy is maximized when all events are equiprobable, that is:

$$H(p_1, p_2, \dots, p_n) \leq H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) = \log(n) \quad (9)$$

Property 2: If \mathcal{A}_n and \mathcal{B}_m are any two independent finite fields, $\mathcal{A}_n \times \mathcal{B}_m$ a new finite field with nm events, then

$$H(\mathcal{A}_n \times \mathcal{B}_m) = H(\mathcal{A}_n) + H(\mathcal{B}_m) \quad (10)$$

Shannon's entropy is equally applicable to partitioned sets of events. More precisely, given a partitioning $\Pi = \{A_1, A_2, \dots, A_n\}$ on the set of events, the entropy of this partitioning is:

$$H(\Pi) = - \sum_{i=1}^n p(A_i) \log p(A_i) \quad (11)$$

where $p(A_i)$ is the probability of class A_i in partition Π .

The applications of entropy fall in two categories. The first category consists of problems related to the determination of unknown distributions based on the principle of maximum entropy; that is, determining the distribution of some partition of events subject to given constraints (statistical mechanics). In the second category (coding theory), we are given the source entropy and wish to construct various random variables (code lengths) so as to minimize their expected values. In this paper, we focus on the relationship between the information (entropy) contained in a binary stream and the resulting power consumption when this binary stream is applied to the inputs of a combinational logic block.

Example 1: The truth table for a randomly excited 1-bit full adder is given below:

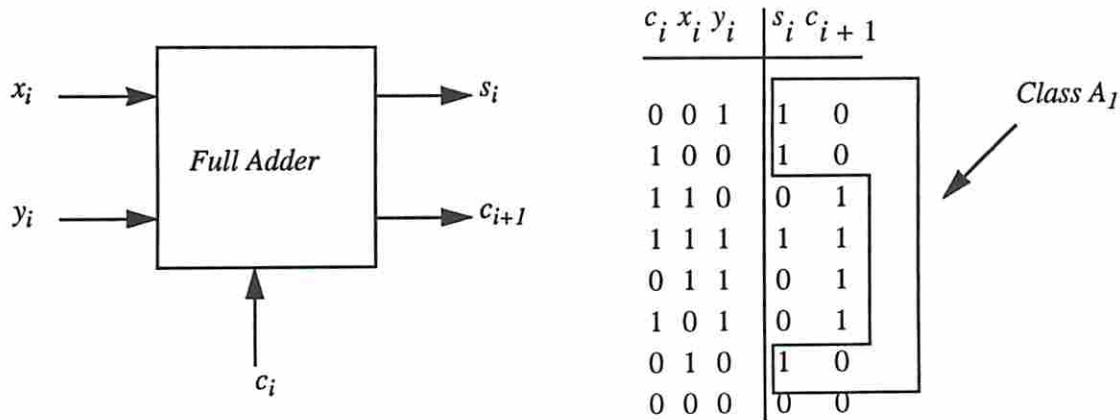


Fig.5: 1-bit full adder

where x_i , y_i are the inputs, c_i is carry-in, s_i is the sum bit and c_{i+1} is carry-out. The output space is partitioned in four classes as $\Pi = \{A_1, A_2, A_3, A_4\} = \{10, 01, 11, 00\}$, where $p(A_1) = p(A_2) = 3/8$, $p(A_3) = p(A_4) = 1/8$; applying (11) we obtain $H(\Pi) = 1.8113$. We observe that within a class there is no activity on the outputs; this means that output transitions may occur only when one has to cross class boundaries in different time steps. If the output sequence is a purely random one, then exactly H bits are needed to represent the output sequence; therefore the average number of transitions per word (or average switching activity per word) will be $H/2$. In any other nonrandom arrangement, for a minimum length encoding scheme, the average number of transitions per word will be $\leq H/2$, so in practice, $H/2$ can serve as a conservative upper bound on the number of transitions per word. In our example, we find an average switching value approximately equal to 0.905 which matches fairly well the exact value 1 deduced from the above table. Such a measure was suggested initially by Hellerman to quantify the *computational work* of simple processes [24]; it was subsequently explored using symbolic conditional expressions for the output probabilities [30].

More formally, if a signal x is modelled as a lag-one Markov chain with conditional probabilities p_{00} , p_{01} , p_{10} , p_{11} and signal probabilities p_0 and p_1 as in Fig.6, then we can characterize it through the conditional

entropy between 2 successive steps in time as: $h(x^+|x^-) = -p_1 (p_{10} \log p_{10} + p_{11} \log p_{11}) - p_0 (p_{00} \log p_{00} + p_{01} \log p_{01})$.

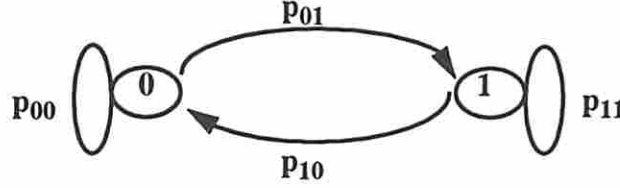


Fig.6: A two-state Markov-chain modelling a signal line x

The signal probabilities can be expressed in terms of conditional probabilities as: $p_1 = \frac{P_{01}}{P_{01} + P_{10}}$ and

$$p_0 = \frac{P_{10}}{P_{01} + P_{10}} \text{ respectively [16]. Using the well-known identity } -\ln(1-a) = \sum_{k=1}^{\infty} \frac{a^k}{k} \text{ for } 0 < a < 1, \text{ we}$$

obtain:

$$h(x^+|x^-) = \frac{1}{2 \cdot \ln(2)} \cdot \frac{2 \cdot P_{01} \cdot P_{10}}{P_{01} + P_{10}} \cdot \sum_{k=1}^{\infty} \left[\frac{\left(\frac{P_{00}^{k-1} + P_{01}^{k-1}}{P_{01} + P_{10}} \right) \cdot P_{00} + \left(\frac{P_{10}^{k-1} + P_{11}^{k-1}}{P_{01} + P_{10}} \right) \cdot P_{11}}{k} \right].$$

We note that $\frac{2 \cdot P_{01} \cdot P_{10}}{P_{01} + P_{10}}$ is exactly the switching activity of line x (denoted by $sw(x)$) and that $a^{k-1} + b^{k-1}$ (for

$$a + b = 1) \text{ is minimum when } a = b = 0.5. \text{ Thus, } h(x^+|x^-) \geq \frac{1}{2 \cdot \ln(2)} \cdot sw(x) \cdot \sum_{k=1}^{\infty} \frac{(p_{00} + p_{11}) \cdot \frac{1}{2^{k-2}}}{k} \text{ or,}$$

using again the above identity: $h(x^+|x^-) \geq 2 \cdot sw(x) \cdot (p_{00} + p_{11})$. So, what we got is an upper bound for the switching activity of signal x in the most general case when it is modelled as a lag-one Markov chain: $sw(x) \leq h(x^+|x^-) / [2 \cdot (p_{00} + p_{11})]$. Unfortunately, there is no relationship between p_{00} and p_{11} and their sum can take any value. To obtain an upper bound useful in practice, we assume *temporal independence*, that is: $p_1 = p_{01} = p_{11}$, $p_0 = p_{10} = p_{00}$ and $h(x^+|x^-) = h(x)$. In this simpler case, $p_{00} + p_{11} = 1$ and hence the relationship between $sw(x)$ and $h(x)$ is exactly the one based on intuition: $sw(x) \leq h(x) / 2$.

To evaluate H , one can use basic results from information theory concerning transmission of information through a module. More precisely, for a module with input X and output Y , we have $I(X;Y) = H(X) - H(X|Y)$ and by symmetry $I(X;Y) = H(Y) - H(Y|X)$ due to the commutativity property of mutual information. When input X is known, no uncertainty is left about the output Y and thus $H(Y|X)$ is zero. Therefore, the information transmitted through a module can be expressed as

$H(Y) = H(X) - H(X|Y)$ which represents the amount of information provided about X by Y . For instance, in Fig.5, $I(X; Y) = H(Y) = 3 - 1.1887 = 1.8113$; thus, informally at least, the observation of the output of the module provides 1.8113 bits of information about the input, on average. However, in real examples, this type of characterization is essentially very expensive as long as the input/output relation is not a one-to-one mapping. This usually ends up in intricate analytical calculations; for instance, the exact calculation of the output entropy of an n -bit adder, would require the knowledge of joint input/output probabilities and a double summation with 2^{2n} terms (as in (6)).

For interconnected modules, the estimation of output entropy for the entire design is even more complicated. Assume the following interconnection structure among n -bit adders (Fig.7). To find out the information transmitted through the whole design, we are faced with very expensive computations: the number of needed joint probabilities (and the number of terms in the summation (6)) becomes $\Theta(2^{4n})$. Therefore, this approach is quite impractical.

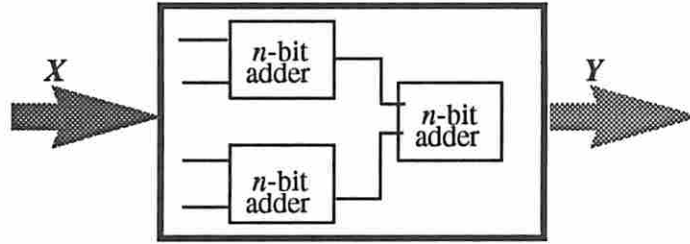


Fig.7: Complexity increase for interconnected modules

As a consequence, in order to analyze large designs, we target a *compositional approach* where the basic modules are already characterized in terms of transinformation and what is left to find, is only a *propagation mechanism* among them (all details are given in Section IV).

As we have seen, an appropriate measure for the average switching activity of each net in the circuit is its entropy value. Basically, what we need is a mapping $\xi \rightarrow \xi'$ from the actual set ξ of the nets in the target circuit (each having a possibly distinct switching activity value) to a virtual set ξ' , which contains the same collection of wires, but this time each net has the same value of switching activity. More formally, $\xi \rightarrow \xi'$ is a mapping such that the following conditions are satisfied:

$$\begin{cases} |\xi| = |\xi'| \\ sw(x_i) = sw(x_j) \quad x_i, x_j \in \xi' \end{cases} \quad (12)$$

Bearing in mind this, one can express the total number of transitions per step as:

$$SW(\xi) = SW(\xi') \leq n \cdot \frac{h(\xi')}{2} \quad (13)$$

where n stands for the presumed cardinality of ξ' and $h(\xi')$ represents the average entropy per bit of any net in ξ' . To clarify these ideas, let us consider the simple circuit in Fig.8.

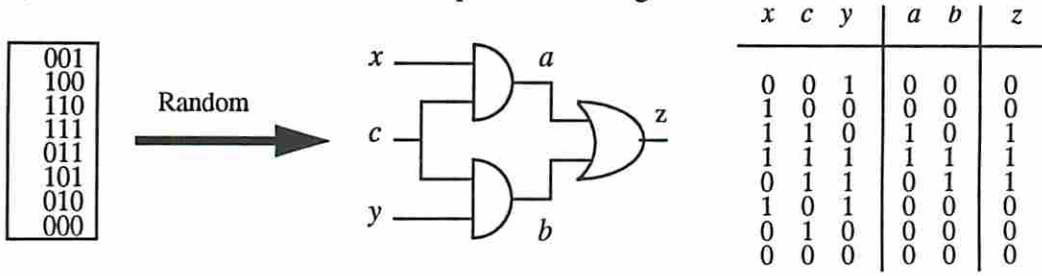


Fig.8: An example to illustrate the mapping $\xi \rightarrow \xi'$

In this example, we feed the circuit with a 3-bit random sequence and tabulate in the right side, the logic values obtained in the entire circuit by logic simulation. We have that $\xi = \{x, c, y, a, b, z\}$ with the switching profile (in number of transitions) $\{4, 4, 4, 2, 2, 2\}$, respectively. Doing a quick calculation, we get $SW(\xi) = 2.25$ transitions per step. On the other hand, $\xi' = \{x, c, y, a, b, z\}$ with the average entropy $h(\xi') = (3 * 1 + 2 * 0.811 + 0.954) / 6 = 0.9295$ characterizing each signal in the set; using relation (13) we get an expected value $SW(\xi') = 2.73$ which is greater than $SW(\xi)$, but sufficiently close to it.

Unfortunately, in large circuits, it is expensive to compute $h(\xi')$ as the complete set of events characterizing a circuit is exponential in the number of nodes. To avoid the brute force approach, that is exhaustive logic simulation, we make some simplifying assumptions which will be detailed in Section III.A.

B. An Informational Energy-Based Approach

Assuming that we have a complete set of events \mathcal{A}_n (as in (3)), we may regard the probability p_k as the *information* associated with the individual event A_k . Hence, as in the case of self-information, we may define an average measure for the set \mathcal{A}_n :

Definition 5: (Informational Energy)

The global information of the finite field \mathcal{A}_n (denoted by $E(\mathcal{A}_n)$) may be expressed by its *informational energy* as (also called the *Gini Function*¹):

$$E(\mathcal{A}_n) = \sum_{j=1}^n p_j^2 \quad (14)$$

We plot in Fig.9 the informational energy of a Boolean variable as a function of its signal probability, that is

$$E(\mathcal{A}_2) = p^2 + (1-p)^2.$$

¹This was first used by Corrado Gini in a study from "Atti del R. Ist. Veneta di Scienze," *Lettere ed Arti*, 1917, 1918, v. LXXVII

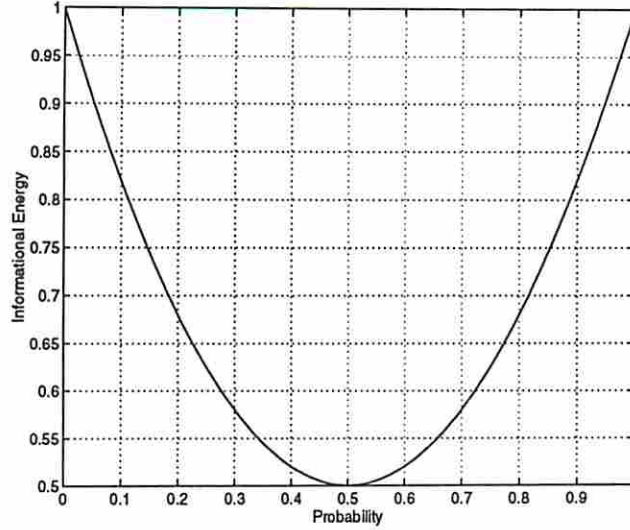


Fig.9: Informational energy $E(\mathcal{A}_2)$ vs. probability

Due to its simplicity, the informational energy was used mainly in statistics (not necessarily in conjunction with Shannon's entropy) as a characteristic of a distribution (discrete as is the case here, or continuous in general). However, without a precise theory, its use was rare until its usefulness was proved [31].

The value of the informational energy is always upper-bounded by 1. This is because

$$\sum_{j=1}^n p_j^2 \leq \left(\sum_{j=1}^n p_j \right)^2 = 1 \text{ with equality iff one of the events has probability 1 and the rest of them have the}$$

probability 0. Thus, $E(\mathcal{A}_n) = 1$ iff the experiment provides the same determinate and unique result. We give in the following few basic properties satisfied by the informational energy [32].

Property 3: The informational energy becomes $1/n$ when all events are equally likely and 1 when one of the events in \mathcal{A}_n is certain:

$$E\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) = \frac{1}{n} \leq E(\mathcal{A}_n) \tag{15}$$

Property 4: If the uniformity (or the uncertainty) of the system increases, then its informational energy decreases.

Property 5: If \mathcal{A}_n and \mathcal{B}_m are two independent finite fields, then $E(\mathcal{A}_n \times \mathcal{B}_m) = E(\mathcal{A}_n) \cdot E(\mathcal{B}_m)$.

Based on this measure, one can find a relationship between the switching activity and the informational energy. Let $e(x)$ denote the informational energy of a single bit x . Considering x modelled as a lag-one Markov chain with conditional probabilities $p_{00}, p_{01}, p_{10}, p_{11}$ and signal probabilities p_0 and p_1 as in Fig.6, we can characterize it through the conditional informational energy between two successive steps in time by: $e(x^+|x^-)$

$= p_1 (p_{10}^2 + p_{11}^2) - p_0 (p_{00}^2 + p_{01}^2)$. Since the switching activity of line x is $\frac{2 \cdot p_{01} \cdot p_{10}}{p_{01} + p_{10}}$ and $p_1 = \frac{p_{01}}{p_{01} + p_{10}}$,

$p_0 = \frac{p_{10}}{p_{01} + p_{10}}$ one can write the following relationship between conditional informational energy and

switching activity: $e(x^+|x^-) = 1 - sw(x) \cdot (p_{00} + p_{11})$. Again, in the most general case, p_{00} and p_{11} can take any values and thus even if we have an exact relation between energy and switching activity ($sw(x) = (1 - e(x^+|x^-)) / (p_{00} + p_{11})$) we cannot bound the sum from the denominator. However, under temporal independence assumption [16], we have an *exact* relation since $p_{00} + p_{11} = 1$ and thus:

$$sw(x) = 2p(x)(1 - p(x)) = 1 - e(x) \quad (16)$$

For instance, returning to the 1-bit full-adder example in Fig.5, we find $E_{input} = 0.125$ and $E_{output} = 0.875$. Thus, on average, the output exposes an informational energy of 0.437 and, based on (16), a switching activity of 0.563 (compared to the exact value of 0.5). Thus, informational energy (along with entropy) seems to be a reliable candidate for estimation of energy consumption.

Once again we consider the virtual mapping $\xi \rightarrow \xi'$, where each net in ξ' is characterized by the same amount of average informational energy $e(\xi')$. Based on (12), the expected switching activity per step in the whole circuit $SW(\xi')$, can be expressed as:

$$SW(\xi) = SW(\xi') = n \cdot (1 - e(\xi')) \quad (17)$$

where the cardinality of ξ' is assumed to be n .

Considering the simple case in Fig.8, we get $e(\xi') = (3 \cdot 0.5 + 2 \cdot 0.625 + 0.531) / 6 = 0.546$ and therefore, $SW(\xi') = 2.71$ which matches well the actual value (2.25).

However, in real circuits, direct computation of $e(\xi')$ is very costly; to develop a practical approach, we need further simplifications as will be shown subsequently.

C. Quantitative Evaluations

In order to derive a consistent model for energy consumption at RT-level, we first have to abstract somehow the information present at gate level. Thus, a simplified model is taken as a starting point.

Let us consider some combinational block realized on n levels as a leaf-DAG¹ of 2-input NAND gates (a similar analysis can be carried out for 2-input NOR gates and the final result is the same). We assume that inverters may appear only at primary inputs/outputs of the circuit; we do not include these inverters in the level assignment step. One can express the signal probability of any net at level $j+1$ as a function of the signal

¹In a leaf-DAG, only the leaf nodes have multiple fanouts.

probability at level j by:

$$p_{j+1} = 1 - p_j^2 \quad \forall j = 0, \dots, n-1 \quad (18)$$

Similarly, the signal probability of any net at level $j+2$ is given by:

$$p_{j+2} = 1 - \left(1 - p_j^2\right)^2 \quad \forall j = 0, \dots, n-2 \quad (19)$$

The average entropy per net at level j is given by:

$$h_j = -p_j \cdot \log p_j - (1 - p_j) \cdot \log (1 - p_j) \quad (20)$$

Using the corresponding average entropy per net at level $j+2$, the parametrized relationship between h_j and h_{j+2}

can be approximated by $h_{j+2} \approx \frac{h_j}{2}$ when j is sufficiently large (values greater than 6). Hence we get

expressions for entropy per bit at even/odd levels of the circuit: $h_{2j} \approx \frac{h_0}{2^j}$ and $h_{2j+1} \approx \frac{h_1}{2^j}$, where h_0, h_1 are

entropies per bit at the primary inputs and first level, respectively. To get a closed form expression, we may

further assume that h_1 may be estimated in terms of h_0 as $h_1 \approx \frac{h_0}{\sqrt{2}}$ (in fact, the exact entropy decrease for a

pseudorandom excited NAND gate is 0.811, but for uniformity, we chose this way). Thus, for a 2-input NAND gate leaf-DAG, the entropy per bit at level j may be approximated as:

$$h_j \approx \frac{h_0}{2^{j/2}} \quad (21)$$

This may be further generalized for the case of f -input NAND gate leaf-DAGs, observing that increasing the fanin from 2 to f , produces an decrease in the number of levels by $\log(f)$. Hence, for a fanin of f , (21) becomes:

$$h_j \approx \frac{h_0}{2^{(j \cdot \log f)/2}} = \frac{h_0}{f^{j/2}} \quad (22)$$

We call $\frac{1}{\sqrt{f}}$ *information scaling factor*; it characterizes each logic component (gate, module or circuit). We

will see how this relation is affected by the circuit structure and functionality in general. In any case, this provides a starting point for estimating the total entropy at each level in the circuit. In general, the total entropy over all levels N in the circuit would thus be:

$$H_{total} = \sum_{j=0}^N H_j = \sum_{j=0}^N n_j \cdot h_j \quad (23)$$

where H_j is the total entropy at level j and n_j is the number of nodes on level j .

All these considerations can be easily extended for the case of informational energy. Considering the same assumptions as in previous section and using relation (18), the informational energy per net at level j may be expressed as:

$$e_j = p_j^2 + (1 - p_j)^2 \quad (24)$$

Applying (18) for level $j+2$ and substituting in (24), we get the following parameterized dependency between the informational energies at levels $j + 2$ and j :

$$e_{j+2} = \left(1 - \left(1 - p_j^2\right)^2\right)^2 + \left(1 - p_j^2\right)^4 \quad e_j = p_j^2 + (1 - p_j)^2 \quad (25)$$

Using a similar approach as in the case of entropy, we get the following expression for the average informational energy per bit at level j in a circuit with fanin f :

$$e_j = 1 - \frac{1 - e_0}{f^{j/2}} \quad (26)$$

From here, an estimate can be drawn for the total energy at level j , and thus for the total energy over all the levels of the circuit:

$$E_{total} = \sum_{j=0}^N E_j = \sum_{j=0}^N n_j \cdot e_j \quad (27)$$

where E_j is the total energy at level j and again n_j is the number of nodes on level j .

III. INFORMATION MODELING

A. Theoretical Results

As we have seen, an estimate of the average switching activity for a module can be obtained from the total entropy (informational energy) over all levels of the circuit. An exact technique would be too expensive to use in practice; on the other hand, since we are dealing with RT-level designs, the internal structure may be unknown. So, in order to make things manageable, we will use the following simplifying assumptions:

A₁. Uniform Network Structure: Nodes are uniformly distributed over the levels of the circuit.

In other words, we assume the same number of nodes on each level of the circuit. Also, all the gates on each level are assumed to get their inputs from the previous level. This will significantly simplify our task in obtaining closed-form formulae for average switching activity per module (see Section IV.A for the effect of other common network structures when assumption A_1 is relaxed).

As we have seen, for particular structures (that is leaf-DAGs containing 2-input NAND/NOR gates), there is a simple relationship between the values of entropy (informational energy) on adjacent levels in the circuit. Unfortunately, in practice these structures are too restrictive to be considered alone: random logic circuits exhibit a large fanout, not only at the primary inputs, but also at internal nodes. This is one reason to reconsider the above relations and make them applicable in practice. In addition, logic circuits contain a mixture of gates: while NAND (AND), NOR (OR) are *entropy decreasing*, XORs and inverters are *entropy preserving* gates. More precisely, the output entropy of XORs is 1 when they are randomly excited and therefore their information scaling factor is 1. Their behavior is still described by (22) for $f = 1$ (similar considerations apply to informational energy). In general, any generic “gate” having almost equal-sized ON and OFF sets, exposes the same *almost entropy preserving* characteristic.

Example 2:

<i>a</i>	<i>b</i>	<i>c</i>	<i>f</i>	<i>g</i>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	0

As we can easily see, the boolean function f has equal ON- and OFF-sets and thus, implements an information preserving gate. If exhaustively excited, the output entropy of f is:

$$H_f = -0.5 \log(0.5) - 0.5 \log(0.5) = 1 \text{ and thus the information scaling factor is 1.}$$

On the other hand, g has unbalanced ON- and OFF-sets (only 2 out of 8 minterms are in the ON-set). The corresponding output entropy is:

$$H_g = -0.25 \log(0.25) - 0.75 \log(0.75) = 0.8113 \text{ which corresponds to its information scaling factor.}$$

In short, we may say that both structural and functional aspects are important. At RT-level both of them have to be abstracted and used in an implicit manner to compensate the lack of explicit information which characterize high-level representations. To overcome this difficulty, we will use another simplifying assumption:

A_2 . *Uniform Information Variation:* The entropy and informational energy per bit at level j are estimated in terms of f_{eff} as:

$$h_j = \frac{h_0}{f_{eff}^{j/2}} \quad \text{and} \quad e_j = 1 - \frac{1 - e_0}{f_{eff}^{j/2}}, \quad j = 0, 1, \dots, N. \quad (28)$$

Differently stated, we assume that each “generic gate” from a given circuit is characterized by an *effective information scaling factor* whose value depends on both structure and functionality of gates. In fact, the above formula is similar with (22) and (26) derived for a particular circuit (leaf-DAG).

Under assumptions A_1 and A_2 , we may state the following:

Proposition 1: The average entropy (informational energy) per bit in an N -level circuit, may be estimated as:

$$h_{avg} = h_{in} \cdot \frac{1 - \left(\frac{h_{out}}{h_{in}}\right)^{\frac{N+1}{N}}}{(N+1) \cdot \left(1 - \left(\frac{h_{out}}{h_{in}}\right)^{\frac{1}{N}}\right)} \quad e_{avg} = 1 - (1 - e_{in}) \cdot \frac{1 - \left(\frac{1 - e_{out}}{1 - e_{in}}\right)^{\frac{N+1}{N}}}{(N+1) \cdot \left(1 - \left(\frac{1 - e_{out}}{1 - e_{in}}\right)^{\frac{1}{N}}\right)} \quad (29)$$

where h_{in} (e_{in}), h_{out} (e_{out}) are the average input and output entropies (energies) per bit.

Proof: From (28), the average entropy per bit at level j as a function of input entropy $h_{in} = h_0$: is $h_j = \frac{h_{in}}{f_{eff}^{j/2}}$

Thus, summing-up over all the levels and taking the average we get:

$$h_{avg} = \frac{1}{N+1} \cdot \sum_{j=0}^N \frac{h_{in}}{f_{eff}^{j/2}} = \frac{h_{in}}{N+1} \cdot \sum_{j=0}^N \frac{1}{f_{eff}^{j/2}} = h_{in} \cdot \frac{1 - \left(\frac{1}{\sqrt{f_{eff}}}\right)^{N+1}}{(N+1) \cdot \left(1 - \frac{1}{\sqrt{f_{eff}}}\right)} \quad (30)$$

On the other hand, for $j = N$, relation (28) is applicable to the outputs:

$$h_{out} = \frac{h_{in}}{f_{eff}^{N/2}} \text{ and from here, } \frac{1}{\sqrt{f_{eff}}} = \left(\frac{h_{out}}{h_{in}}\right)^{\frac{1}{N}}. \text{ Plugging this in (30), we get exactly the claim of the}$$

Proposition. Similar considerations apply to informational energy. \square

Proposition 1 gives us an estimate of the average entropy/informational energy in a circuit with N levels. The factor f_{eff} is “hidden” in the relationship between N , h_{in} (e_{in}) and h_{out} (e_{out}) since the outputs are considered to be on level N :

$$h_{out} = h_N = \frac{h_0}{f_{eff}^{N/2}} = \frac{h_{in}}{f_{eff}^{N/2}} \quad e_{out} = e_N = 1 - \frac{1 - e_0}{f_{eff}^{N/2}} = 1 - \frac{1 - e_{in}}{f_{eff}^{N/2}} \quad (31)$$

The greater the number of levels is, the smaller the value for f_{eff} will be, which somehow suggests that the loss of information per bit from one level to another decreases with the number of levels. However, the usefulness of these formulas is limited, since in general at RT-level we know very little about the internal structure of the circuit. To compensate this lack of information, we add a new assumption valid for circuits with large logical depth:

A₃. Asymptotic Network Depth: The number of levels N is large enough to be considered infinity.

Using this, we get the following:

Corollary: For sufficiently large N , the average entropy and informational energy per bit in the circuit are given by:

$$h_{avg} = \frac{h_{in} - h_{out}}{\ln \frac{h_{in}}{h_{out}}} \quad e_{avg} = 1 - \frac{e_{in} - e_{out}}{\ln \frac{1 - e_{out}}{1 - e_{in}}} \quad (32)$$

Proof: From (30), we have that:

$$h_{avg} = \lim_{N \rightarrow \infty} h_{in} \cdot \frac{1 - \left(\frac{h_{out}}{h_{in}}\right)^{\frac{N+1}{N}}}{(N+1) \cdot \left(1 - \left(\frac{h_{out}}{h_{in}}\right)^{\frac{1}{N}}\right)} = (h_{in} - h_{out}) \cdot \lim_{N \rightarrow \infty} \frac{\frac{1}{N+1}}{1 - \left(\frac{h_{out}}{h_{in}}\right)^{\frac{1}{N}}} = \frac{h_{in} - h_{out}}{\ln \left(\frac{h_{in}}{h_{out}}\right)}$$

where we applied the well-known identity $\lim_{x \rightarrow 0} \frac{a^x - 1}{x} = \ln(a)$. A similar calculation can be done for informational energy. \square

Note: In the above derivations, trivial cases such as zero input or output entropy and one input or output energy are excluded.

What we have obtained so far are simple formulae for estimating the average entropy (informational energy) per bit, and from these, the average switching activity over all the nets in the module. The main difficulty in practice is to estimate the actual output entropy h_{out} (or informational energy e_{out}), since the information usually available at this level of abstraction is not detailed.

B. The Influence of Structure and Functionality

All logic gates belonging to a given module can be characterized by an effective factor f_{eff} which captures information about the circuit structure and functionality. How can we model a general circuit for entropy/energy based evaluations? One can consider relations (29) and (31), where the information scaling factor reflects not only the structure, but also the fraction of information preserving gates.

Example 3: Let us consider for instance, circuit C17 given below:

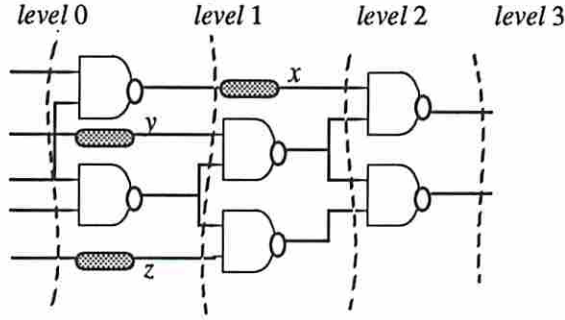


Fig.10: An example of levelization - Circuit C17

To levelize it properly (every wire that connects the output of a gate at level i to the input of a gate at level $i+2$ must go through some buffer gate at level $i+1$), we added three “dummy” components x , y , z . Logically, x , y , z function as buffers but informationally, they are entropy preserving elements. Considering the nodes uniformly distributed throughout the circuit (according to assumption A_1), the average number of nets per level is $(6+5+4+2)/4 = 4.25$. Applying random vectors at the circuit inputs, the exact value of the entropy per bit at the output is obtained as $h_{out} = 0.44$. The effective scaling factor can be calculated as a weighted sum over all

the gates in the circuit; thus the corresponding f_{eff} is:
$$\left(\frac{9}{3 \cdot 1 + 6 \cdot \frac{1}{\sqrt{2}}} \right)^2 = 1.55$$
 (there are 3 entropy

preserving and 6 entropy decreasing gates). From (31) we get an estimate for the output bit entropy ($j = 3$) as $h_{out} = 0.51$ which is reasonably close to the exact value. Based on the input and output entropy, we may get an estimate for the average entropy per bit and thus for the switching activity. The average switching activity for a generic net in the circuit is $sw_{avg(sim)} = 0.437$ (from simulation) and based on (29), we get $sw_{avg(est)} = 0.382$ which is very good compared to simulation. A similar analysis can be performed for the informational energy.

IV. PRACTICAL CONSIDERATIONS

A. Using Structural Information

These considerations are equally applicable to data-path operators with known internal structure as well as to control circuits represented either at gate or RT-level.

If some structural information is available (such as the number of internal nodes, the number of logic levels), the average entropy (informational energy) may be evaluated using the actual values of f_{eff} , N and the distribution of nodes on each level. In all cases, the output entropy (informational energy) is the same, computed as in (28). The average entropy (or informational energy) for the whole module depends on the actual distribution of nodes in the circuit. In practice, some common distributions are:

- a) Uniform distribution (this case was treated in detail in Section III).
 b) Linear distribution (e.g. circuit C17 in Fig.10):

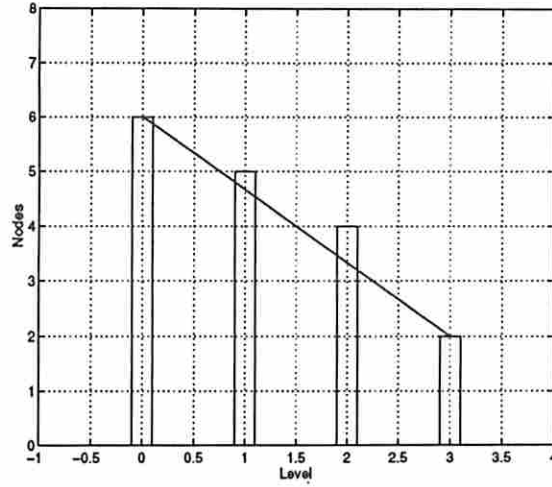


Fig. 11: An example of linear distribution of nodes

Using a similar approach as the one in Section III, we found the following result (valid for sufficiently large N) for a generic n input, m output module:

$$h_{avg} = \frac{2 \cdot n \cdot h_{in}}{(n + m) \cdot \ln\left(\frac{h_{in}}{h_{out}}\right)} \cdot \left(1 - \frac{m}{n} \cdot \frac{h_{out}}{h_{in}} - \frac{\left(1 - \frac{m}{n}\right) \cdot \left(1 - \frac{h_{out}}{h_{in}}\right)}{\ln\left(\frac{h_{in}}{h_{out}}\right)} \right) \quad (33)$$

- c) Exponential distribution (e.g. a balanced tree circuit with 8 inputs):

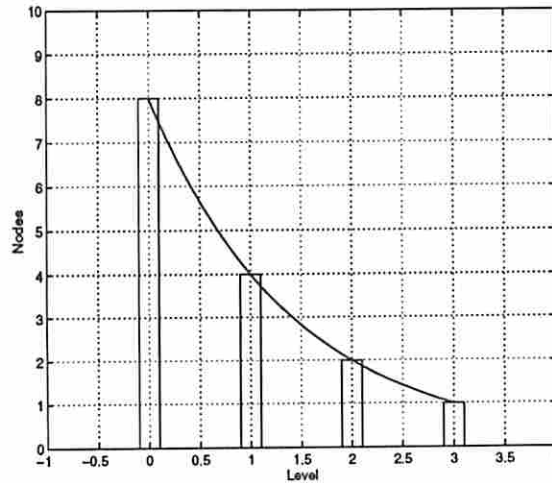


Fig. 12: An example of exponential distribution of nodes

In this case, we have:

$$h_{avg} = \frac{h_{in} \cdot \ln\left(\frac{n}{m}\right)}{1 - \frac{m}{n}} \cdot \frac{1 - \frac{m \cdot h_{out}}{n \cdot h_{in}}}{\ln\left(\frac{n \cdot h_{in}}{m \cdot h_{out}}\right)} \quad (34)$$

Note: The main advantage of relationships (33), (34) is that they allow an estimate of average entropy (and therefore for average switching activity) of modules or gate-level circuits, without resorting to logic simulation or probabilistic techniques like those presented in [3]-[17].

Similar derivations apply for informational energy. We can see that when $n = m$, we get the same results as in Section III (see equation (32)).

B. Using Functional Information

Common data-path operators allow a quick estimation of h_{out} based on the “compositional technique” introduced in [28]. There, the Information Transmission Coefficient (*ITC*) is defined as the fraction of information that is transmitted through a function; it may be computed by taking the ratio of the entropy on the outputs of a function and the entropy on the inputs of that function. For convenience, we call *ITCs* “Entropy Transmission Coefficients” (*HTCs*) either for input or component values. In Table 1 we give the *HTC* values for some common 8-bit data-path operators as they appear in [28].

Table 1: *HTC* Values for Common 8-bit Data-path Operators

Operator	<i>HTC</i>	Operator	<i>HTC</i>
Addition	0.500	Negation	1.000
Subtraction	0.500	And, Or	0.406
Multiplication	0.461	<, >	0.063
Divide by 2	0.875	Multiplexer	0.471

Using the following relationship between the *HTCs* on the output signals and the *HTC* values on the input signals for a particular component, we may estimate the *HTC* values throughout the circuit [28] as:

$$HTC_{out} = HTC_{comp} \cdot \sum_{i=1}^n \frac{w_i}{W} \cdot HTC_i \quad (35)$$

where HTC_{comp} is the *HTC* for the component of interest, HTC_i is the *HTC* value for input i , n is the number of

input signal paths for the component, w_i is the data-path width for input i , and $W = \sum_{i=1}^n w_i$.

In particular, the output entropy h_{out} may be estimated relying solely on the RT-level description of the circuit. The main advantage of such an approach, is that *it needs only a high-level view of the design* in order to derive

useful information.

The HTC value on the inputs is in fact the entropy normalized to the data-path width. To make the compositional technique useful for our purpose, we need to consider both the input and output entropies normalized to the data-path width. For an n -input data-path operator, we have:

$$h_{out} = HTC_{comp} \cdot (n \cdot h_{in}) \quad (36)$$

where h_{in} is the average input bit entropy.

Example 4: Let's apply the compositional technique to the following circuit composed by two 8-bit adders (A_1 and A_2), and one multiplier (M_1).

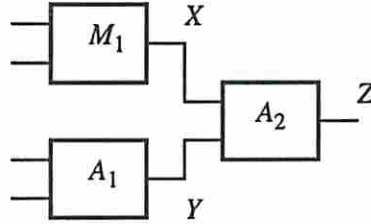


Fig.13: A simple example to illustrate the application of the compositional technique

Since M_1 and A_1 are fed by the primary inputs, we have that $HTC_X = HTC_{M1} = 0.461$ and $HTC_Y = HTC_{A1} = 0.500$. For all the primary inputs of the modules M_1 and A_1 we have $h_{in} = 1$. Accordingly to (36) we have that $h_X = 0.461 \cdot 2 \cdot 1 = 0.92$ and $h_Y = 0.500 \cdot 2 \cdot 1$. Assuming that all paths have the same width, then the weighted

average of the inputs to A_2 is $\frac{h_X + h_Y}{2}$, that is $\frac{1 + 0.92}{2} = 0.96$ and therefore

$$h_Z = 0.500 \cdot 2 \cdot 0.96 = 0.96.$$

However, in general, the $HTCs$ associated with a component are not constant and depend on the input bit entropy (or input $HTCs$) as we shall see in Section IV.C.

A similar technique can be introduced to compute the output informational energy as follows.

Definition 6: The fraction of informational energy transmitted through a function called “Energy Transmission Coefficient” (ETC) is defined as the ratio of the output and input informational energy.

In Table 2 we give the values of the ETC coefficients for the very same data-path operators considered in Table 1.

Table 2: ETC Values for Common Data-path Operators

Operator	ETC	Operator	ETC
Addition	0.500	Negation	1.000
Subtraction	0.500	And, Or	0.625
Multiplication	0.516	<, >	0.063

Table 2: *ETC* Values for Common Data-path Operators

Operator	<i>ETC</i>	Operator	<i>ETC</i>
Divide by 2	1.125	Multiplexer	0.471

Hence, a compositional technique can also be developed here. Similar to (35), we may evaluate ETC_{out} as a function of ETC for the component of interest and the ETC values for all inputs:

$$ETC_{out} = ETC_{comp} \cdot \sum_{i=1}^n \frac{w_i}{W} \cdot ETC_i \quad (37)$$

Thus, for modules with n inputs, the output informational energy is related to the input energy by the following relation:

$$e_{out} = ETC_{comp} \cdot (n \cdot e_{in}) \quad (38)$$

where e_{in} is the input informational energy per bit.

This type of considerations can be used for any type of data-path operator which has the *scalability property* that is, all sizes of the operator behave similarly up to a multiplicative factor (which is the data-path width). Common arithmetic operators exhibit the scalability property but unfortunately, there are many other circuits (e.g. control circuits) which cannot be treated in this manner. In those cases, relation (32) has to be used in conjunction with some information about the circuit structure in order to get reliable estimates for average switching activity.

C. *HTC and ETC Variations with the Input Statistics*

As presented in [28], Thearling and Abraham's compositional techniques is only an approximation because it does not consider any dependency which may arise in practical examples. In reality, every module may be embedded in a larger design and therefore its inputs are no longer independent due to the structural dependencies (namely, the reconvergent fan-out). As a consequence, the values given in Tables 1 or 2 (which correspond to the case of pseudorandom inputs) cannot be used without error as we proceed from circuit inputs to circuit outputs; in order to be accurate we need a more detailed analysis as will be described in the following.

Without loss of generality, we restrict ourselves to the case of 8- and 16-bit adders and multipliers and for each of them, we consider two scenarios:

- Each module is fed by biased input generators, that is input entropy (informational energy) per bit varies between 0 and 1 (respectively 0.5 and 1); each such module is separately analyzed.
- Modules are included in a large design with reconvergent fanout branches, so that inputs of the modules cannot be considered independent. The structure of such an example design is the following:

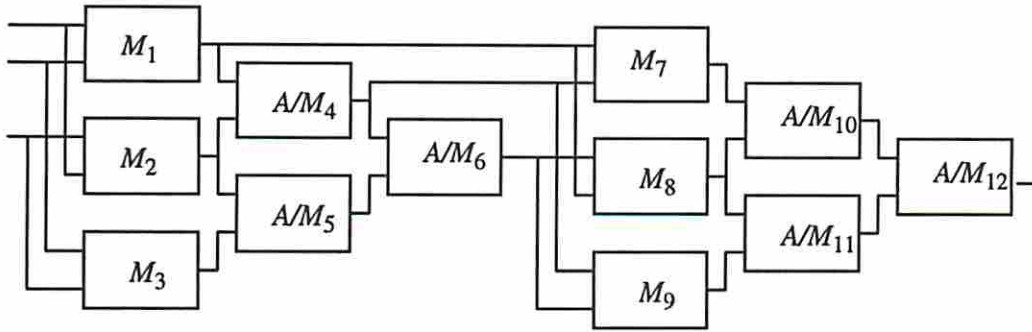


Fig.14: The configuration used to analyze the variation of *HTC/ETC* values

Initially, for the analysis of multipliers, all modules were considered multipliers; in the analysis of adders, modules 4-6, 10-12 were adders (modules 1-3, 7-9 were still kept multipliers to increase the level of correlation in the design).

In both scenarios, the average input and output entropies (energies) per bit were monitored and using them, the *HTC* and *ETC* values were extracted. Values obtained in both scenarios for each module are plotted on the same graph, as shown in Fig.15-18:

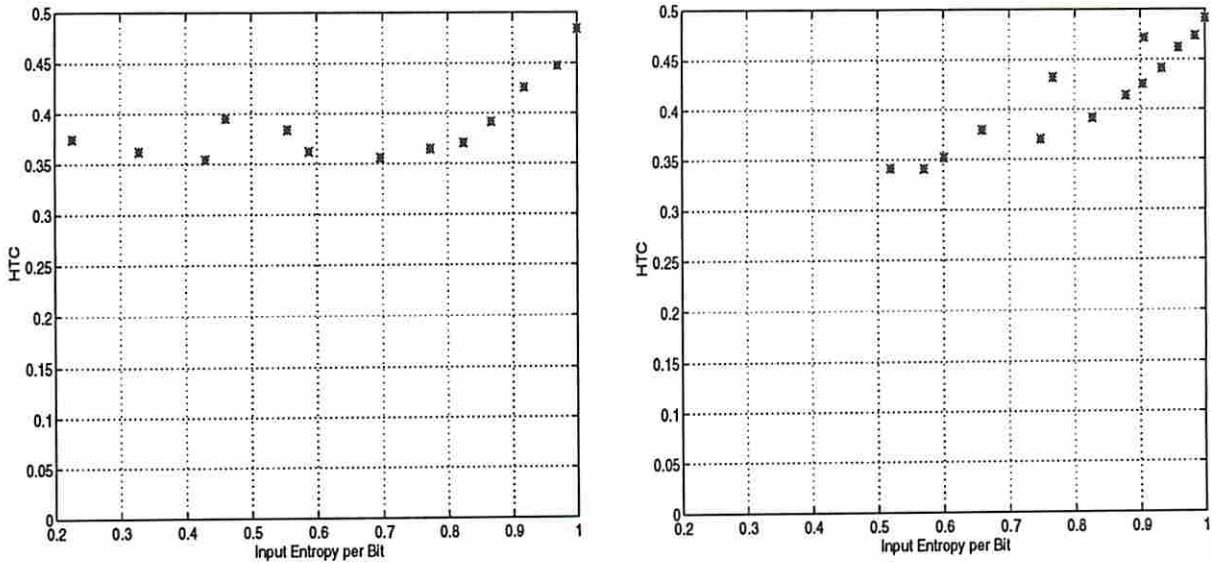


Fig.15: *HTC* values for 8- and 16-bit Multipliers

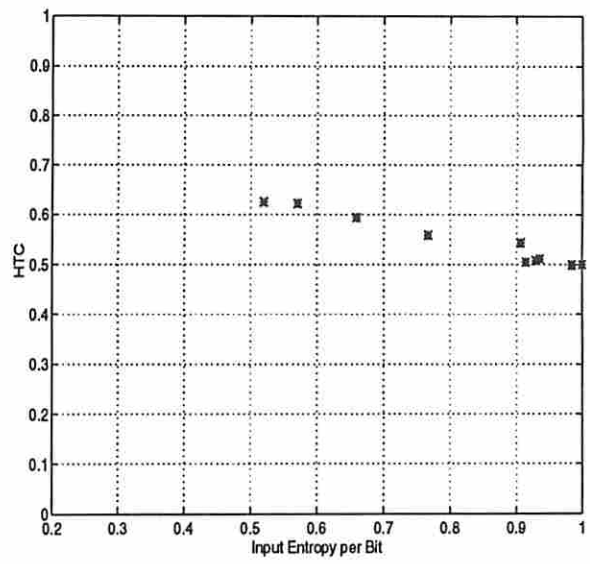
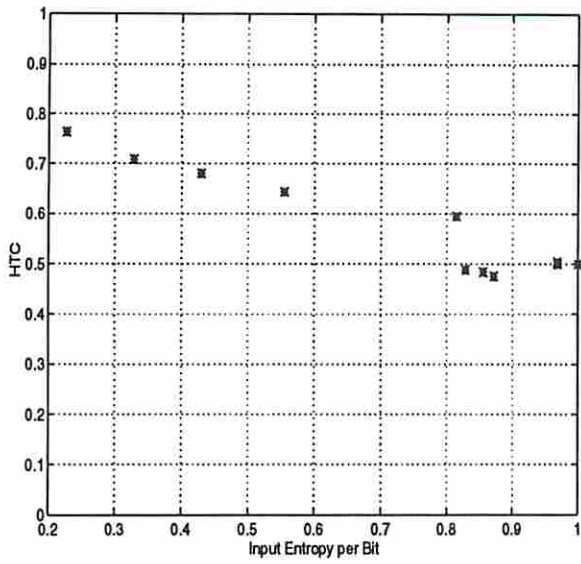


Fig.16: *HTC* values for 8- and 16-bit Adders

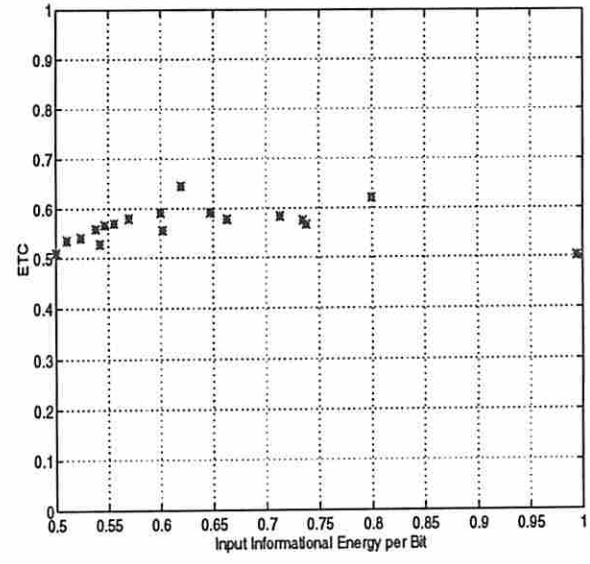
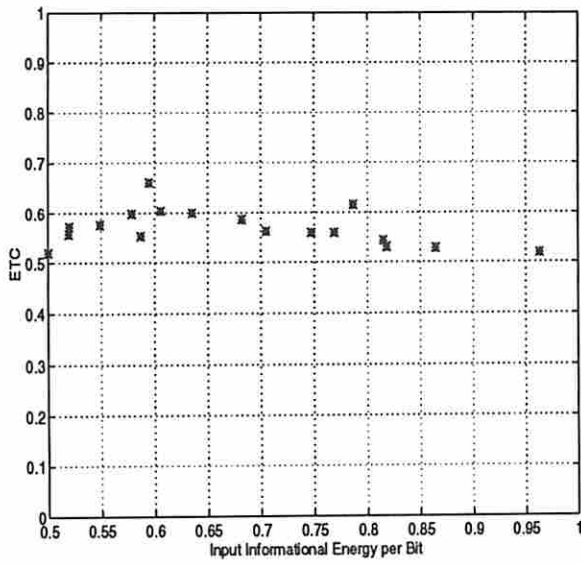


Fig.17: *ETC* values for 8- and 16-bit Multipliers

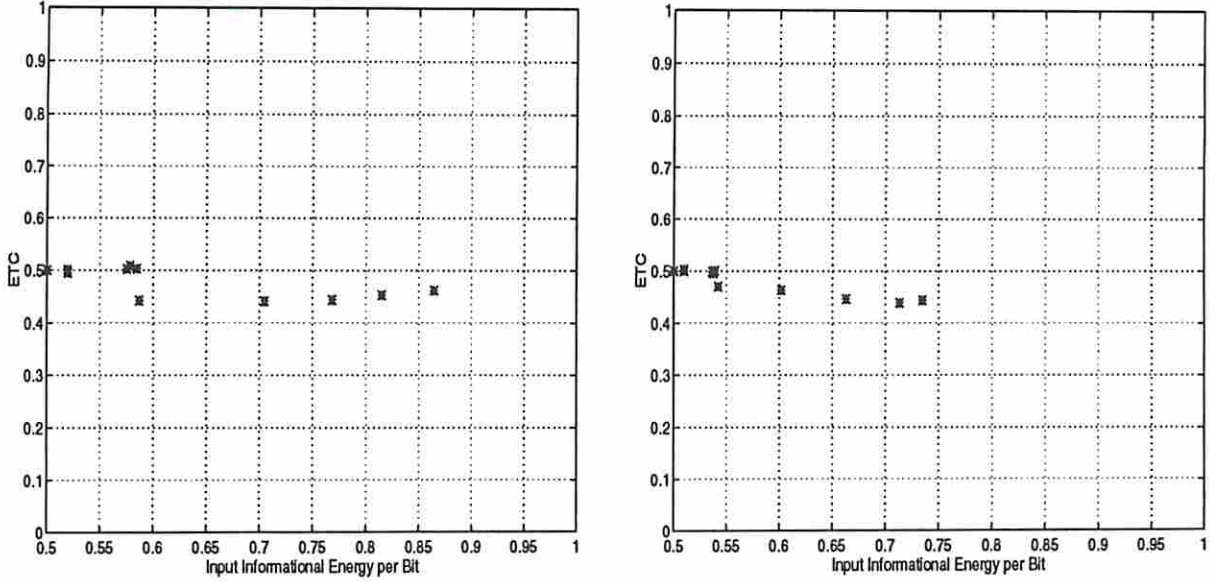


Fig.18: *ETC* values for 8- and 16-bit Adders

As we can see, the behavior of the operators is almost independent of the data-path width, so the same values for *HTC* (*ETC*) can be safely used in both cases. We point out that *ETC* values are less dependent on the input statistics than *HTCs* and presumably, the technique based on informational energy will be less prone to error when the inputs are far from the pseudorandom case. The dependence of *HTCs* and *ETCs* on the input statistics (that is, input entropy and/or informational energy) can be described empirically by the following simple relations:

$$\begin{aligned}
 HTC^{add} &\approx HTC_0^{add} \cdot (2 - h_{in}) & HTC^{mul} &\approx HTC_0^{mul} \cdot 2^{h_{in} - 1} \\
 ETC^{add} &\approx ETC_0^{add} & ETC^{mul} &\approx ETC_0^{mul}
 \end{aligned} \tag{39}$$

where the 0-subscripted values correspond to the pseudorandom case (reported in Tables 3 and 4). These equations can be easily used to adjust the *HTC/ETC* coefficients in order to analyze large designs more accurately. Differently stated, using equations (39) we do not lose information from level to level because we account for structural dependencies.

If the data-path is within a loop, then we may resort to a *loop-unrolling* technique in order to evaluate the output entropy (informational energy). For this purpose, equations (39) can be used to compute the actual values of *HTC/ETC* coefficients across successive iterations. For instance, assume we have to compute the dot-

product of two random vectors X and Y using the following data-flow graph:

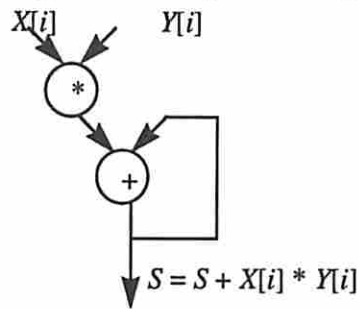


Fig.19: Dot-product computation

Monitoring the output entropy (informational energy) values, we got the following behavior as a function of the number of steps performed in the unrolling process (Fig.20).

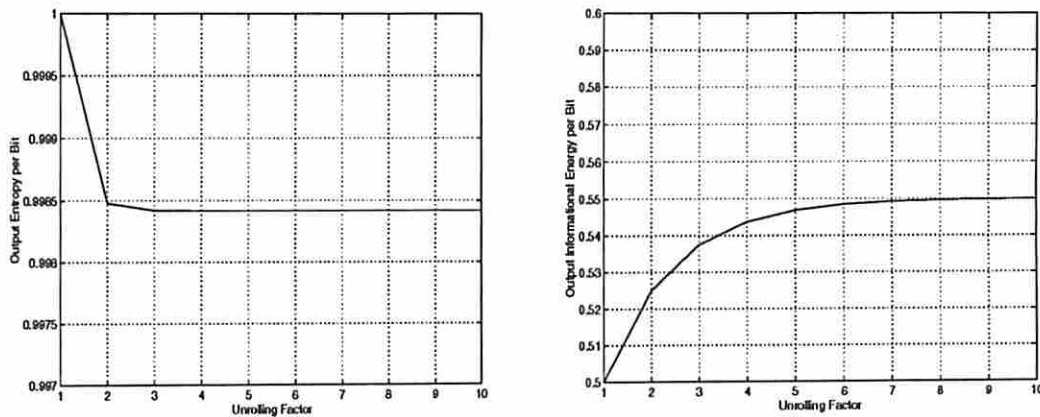


Fig.20: h_{out} and e_{out} behavior during loop-unrolling

This is a typical behavior in practice, that is, the two information measures converge rapidly to the steady-state values and therefore we may exploit this property to use it in conjunction with the compositional technique described before.

In any case, this framework is also open to simulation (the *zero-knowledge scenario*); this may provide accurate values for output entropy (informational energy) values, but with a much higher computational cost.

In practice, we thus have the following options to analyze a logic- or RT-level design:
Less information needed

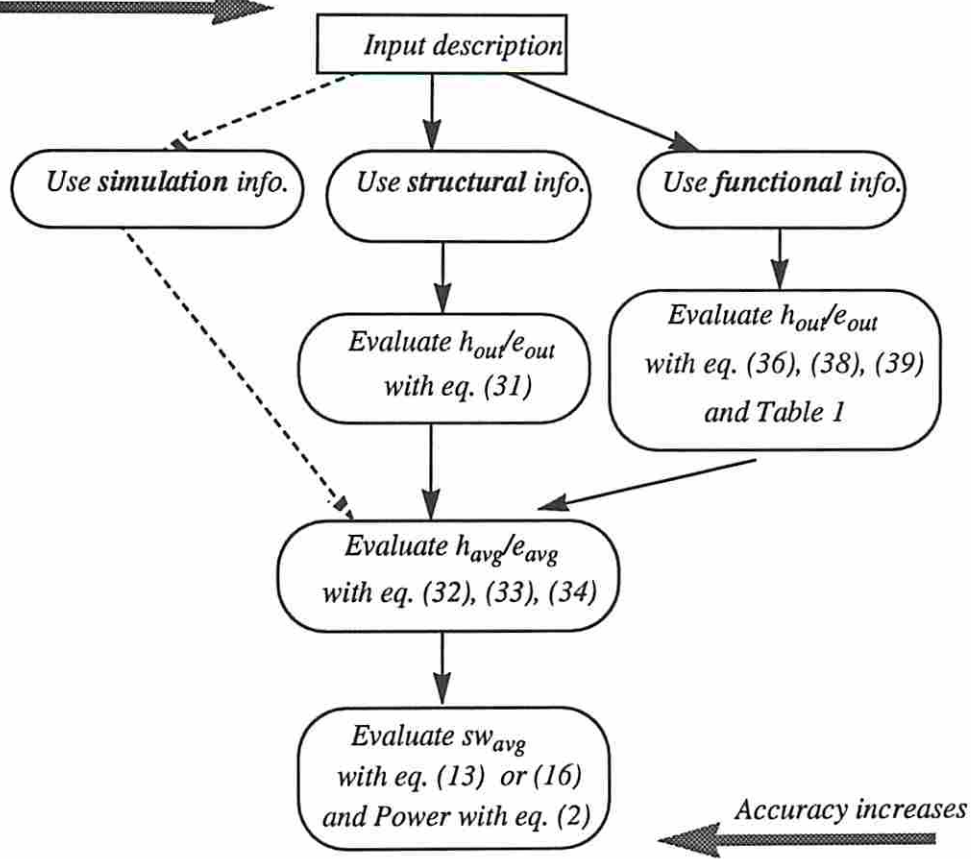


Fig.21: Flowchart of the Power Estimation Procedure

In general, using structural information can provide more accurate results either based on entropy or informational energy measures. On the other hand, evaluations based on functional information require less information about the circuit and therefore may be more appealing in practice as they provide an estimate of power consumption earlier in the design cycle.

To illustrate this aspect, we provide here our results obtained for some common data-path operators (4-bit adders and multipliers) with different structures: ripple-carry and carry-lookahead adders, and array and Wallace-tree multipliers. The exact values for the average switching activity per module are shown in Table 3; they were obtained by exhaustive logic simulation using the SIS logic simulator.

Table 3: Exact Values for Average Switching Activity

Ripple-Carry Adder	Carry-Lookahead Adder	Array Multiplier	Wallace-tree Multiplier
0.4187	0.3591	0.2846	0.2860

Results of Table 3 show that for modules with the same functionality, the structure plays an important role; specifically, the average switching activity may be different even when the input statistics are the same. An

important observation is that looking only at the input/output relationship (even through exhaustive simulation), one is unable to distinguish between two different implementations of the same functional module, but having different average switching activities.

To assess the accuracy of the different approaches in Fig.21, we considered first the two 4-bit adders (ripple-carry and carry-lookahead) having the following node distributions:

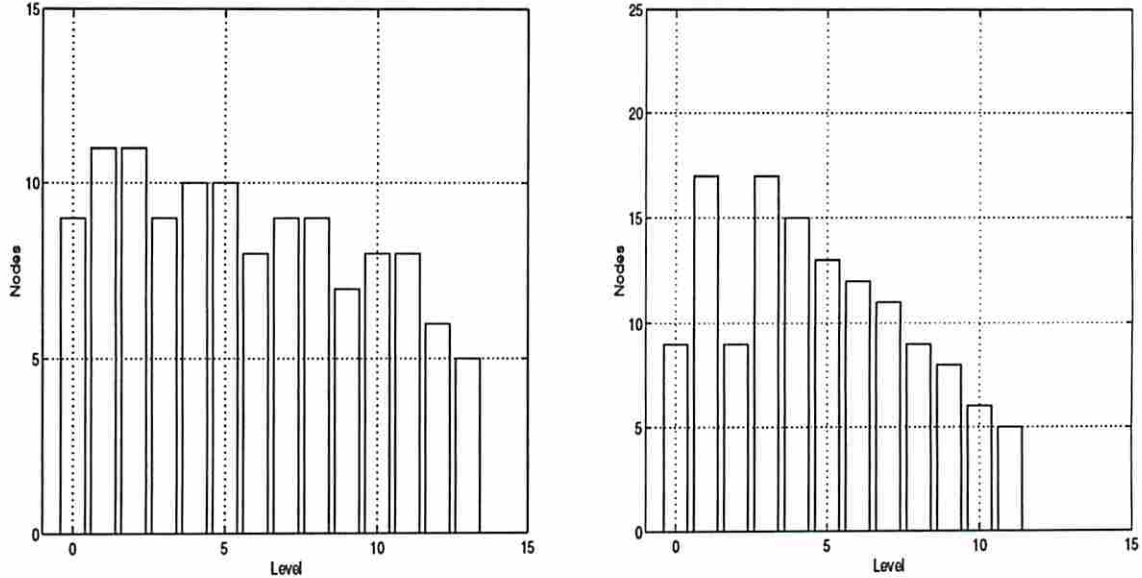


Fig.22: Node distributions for 4-bit ripple-carry (left) and carry-lookahead adder (right)

We estimated first the average switching activity per module using the compositional technique presented previously under the uniform distribution assumption (the default option). After that, assuming that more structural information is available (i.e. node distribution was taken to be linear), we re-calculated the switching activity using equation (31). As shown in Table 4, the quality of results is improved in the latter case. (We report the absolute error defined as $abs_err = |sw_{avg(sim)} - sw_{avg(est)}|$ where $sw_{avg(sim)}$ was taken from Table 3).

Table 4: Analysis of Structurally Different Adders

Measure	<i>abs_err</i> for Ripple-carry Adder		<i>abs_err</i> for Carry-lookahead Adder	
	Functional info	Structural info	Functional info	Structural info
<i>h</i>	0.0494	0.0351	0.1073	0.0588
<i>e</i>	0.0493	0.0350	0.1072	0.0587

The best results are obtained when structural information is available. This is expected as structural information provides more detailed description of the circuit in question.

For the sake of completeness, we also analyzed two structurally different 4-bit multipliers (array and Wallace tree). We give in Fig.23 the node distributions for these circuits and in Table 5 the results of a

complete analysis based on functional or structural information. The distribution of nodes per level was considered exponential for the Wallace-tree multiplier; the actual distribution in the case of array multiplier was approximated by a piecewise linear fit. Again, the best results were obtained when using structural information to calculate the output entropy.

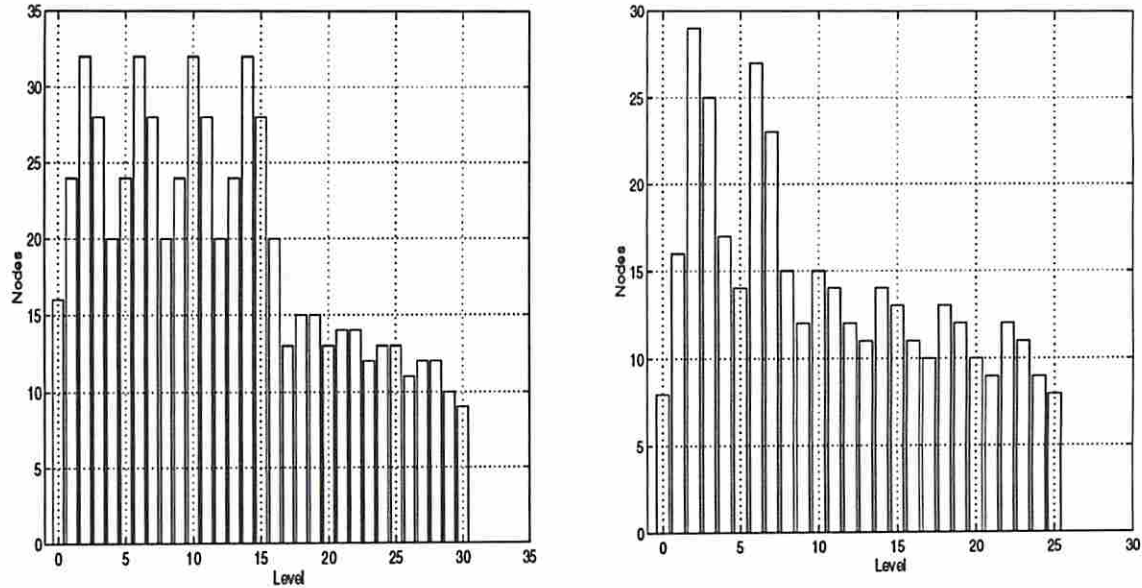


Fig.23: Node distributions for 4-bit array (left) and Wallace-tree multipliers (right)

Table 5: Analysis of Structurally Different Multipliers

Measure	<i>abs_err</i> for Array Multiplier		<i>abs_err</i> for Wallace-Tree Multiplier	
	Functional info	Structural info	Functional info	Structural info
h	0.0517	0.0429	0.0873	0.0102
e	0.0444	0.0427	0.0871	0.0101

To conclude this section, the structural approach is more appropriate to be used when a gate-level description is available (and therefore detailed information can be extracted) whilst the functional approach (using the compositional technique) suits better for RT/behavioral level descriptions.

V. EXPERIMENTAL RESULTS

The main advantage of the proposed power consumption model is that it *does not need any simulation*. In the following we report the accuracy of this model.

Two experiments were performed: one involving individual modules (ISCAS'85 benchmarks and common data-path components) and the other involving a collection of data-path modules specified by a data flow graph.

A. Validation of the Structural Approach (Gate-Level Descriptions)

The experimental setup consisted of a pseudo-random input generator feeding the modules under consideration. The values of the entropy and informational energy for the circuit inputs were extracted from the input sequence, while the corresponding values at the circuit outputs and the average values of entropy or informational energy were estimated as in Section III (using structural information). These average values were then used to estimate the average switching activity per node; the latter, when weighted by an average module capacitance, is a good indicator of power/energy consumption.

We report in Table 6 our results on benchmark circuits and common data-path operators. $Power_{sim}$ and sw_{sim} are the exact values of power and average switching activity obtained through logic simulation under SIS. C_{module} stands for the module capacitance as in (2); it is taken as the product of number of nets and the average load seen by each gate in the circuit. We also report for comparison under the $Power$ column the value of power obtained using the approximation in (2). The error introduced by this approximation is on average 5.55%¹. In the next four columns we report our results for average switching activity and power calculated as in (2) for both entropy- and informational energy-based approaches.

Table 6: Data-Path and ISCAS'85 Circuits (All power values are computed using $V_{DD} = 5V$ and $f_{clk} = 100MHz$)

Simulated values with SIS					Estimated values as in Section III			
Circuit	$Power_{sim}$	C_{module}	sw_{sim}	$Power$	sw_{avg} from h	$Power_{est}$ from h	sw_{avg} from e	$Power_{est}$ from e
add8	1019.39	9.46	0.4199	993.06	0.4410	1042.63	0.4410	1042.51
add16	2082.70	18.91	0.4282	2024.31	0.4546	2149.40	0.4546	2148.58
add32	3980.91	37.82	0.4073	3851.02	0.4923	4654.89	0.4923	4654.74
mul4	2785.53	25.04	0.4170	2610.42	0.4594	2875.60	0.4471	2799.03
mul8	10949.51	100.16	0.4081	10218.82	0.4545	11381.09	0.4410	11043.80
mul16	44076.52	400.64	0.4094	41005.50	0.4515	45222.57	0.4373	43795.33
mul32	166235.89	1602.56	0.3904	156409.86	0.4498	180219.10	0.4351	174333.74
C1355	5163.54	53.59	0.3779	5062.92	0.4261	5707.99	0.4058	5436.64
C1908	5957.64	66.09	0.3417	5645.74	0.4280	7071.99	0.4090	6756.85
C3540	13966.75	175.03	0.2809	12291.48	0.3586	15691.77	0.3281	14354.56
C432	2985.67	29.22	0.3862	2821.19	0.4468	3263.71	0.4331	3163.47
C499	6085.72	61.69	0.3794	5851.30	0.4301	6633.53	0.4120	6354.57
C6288	41860.77	430.24	0.3582	38527.99	0.4354	46832.16	0.4173	44880.26
C880	5200.12	54.78	0.3540	4848.03	0.4465	6115.32	0.4318	5913.98

As we can easily see, the *average percentage error* (over all the circuits) is 15.81% (12.03%) for entropy (informational energy)-based evaluations of average switching activity, whilst for total power estimation is 9.27% (5.85%) for entropy (informational energy)-based approaches..

¹The percentage error was calculated as $\left| \frac{value_{sim} - value_{est}}{value_{sim}} \right| \cdot 100$.

B. Validation of the Functional Approach (Data-Flow Graph Descriptions)

In Fig.24 we consider a complete data-path represented by the data-flow graph of the differential equation example given in [33]; all the primary inputs were considered as having 8 bits and the output entropy of the entire module was estimated with the compositional technique based on *HTCs* or *ETCs*. The *HTC/ETC* values for the multipliers and adders were taken as in (39). Using the entropy based approach, the average switching activity was estimated as 0.1805, whilst when using the informational energy, the average switching activity was 0.1683. Comparing these results against the exact value of 0.1734 obtained by behavioral simulation, the overall error in estimating the average switching activities using entropy and informational energy is 0.0071 and 0.0051, respectively.

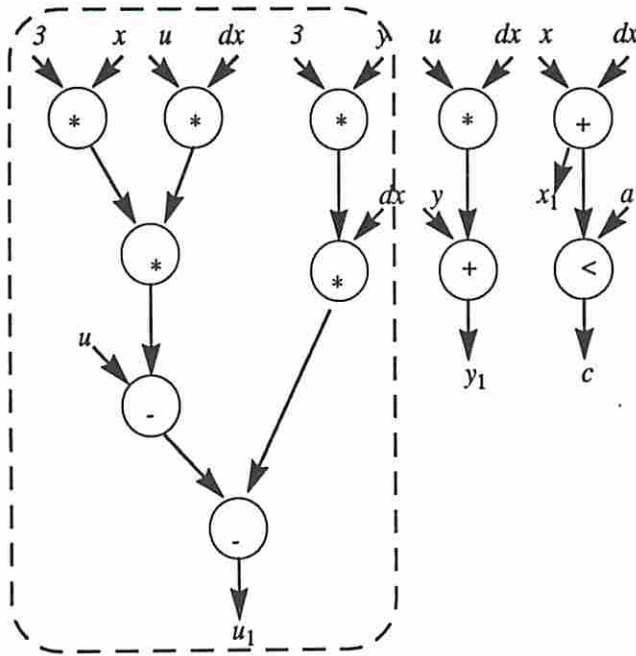


Fig.24: A data-path example

Our technique can also be applied to analyze different implementations of the same design in order to trade-off power for area, speed or testability. Suppose we select from the data-path in Fig.24 only the part which computes u_1 . In Fig.25 we give two possible implementations of the selected part of the data-path. One is the same as above and the other is obtained using common algebraic techniques such as factorization and common subexpression elimination. All the primary inputs were assumed to be random (except inputs '3' and 'dx' which are constants). Each adder or multiplier is labelled with its average switching activity value SW . In the first case, applying the compositional technique based on entropy, we obtain an average switching activity of 0.186, whilst using informational energy this value is 0.197. For the second implementation, the corresponding values are 0.242 from entropy and 0.282 from informational energy which show an average increase in

switching activity of 30%. However, considering the module capacitance of adders and multipliers (as given in Table 6), we actually get a total switched capacitance of 331.15 for the first design and 268.88 for the second one (using entropy-based estimations). This means a decrease of 19%, and thus, the second design seems to be a better choice as far as power consumption is concerned.

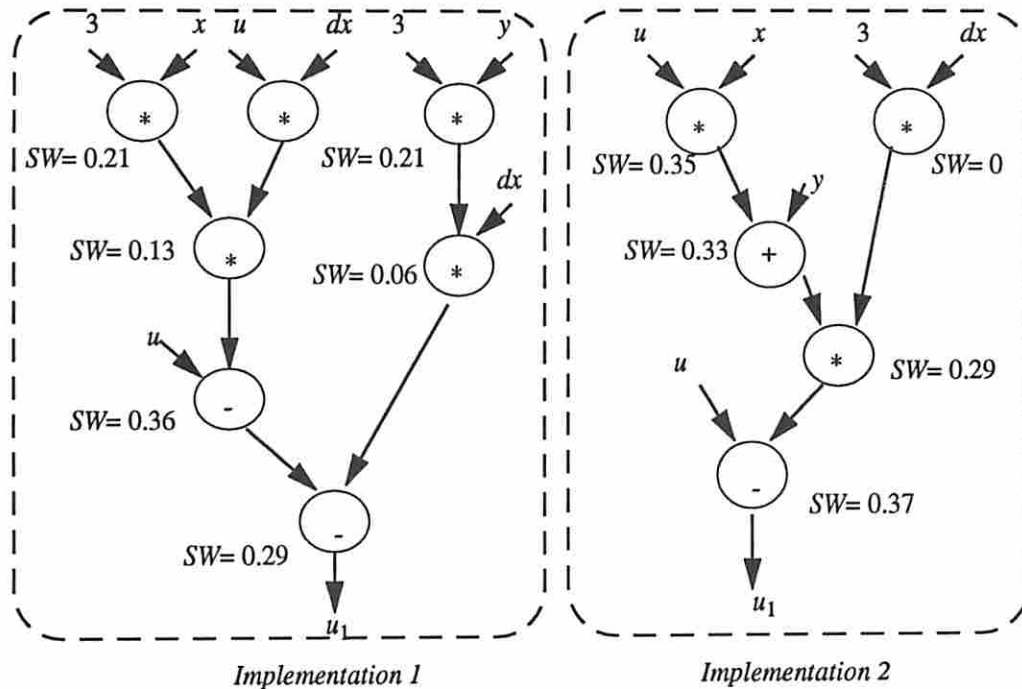


Fig.25: Comparison between two possible implementations

VI. CONCLUSION

In this paper, we have proposed two new measures to estimate the power consumption in logic- or RT-level digital circuits. Their foundation relies on statistical modelling of the circuit behavior and seems to successfully overcome the lack of information which characterizes higher levels of abstraction. The accuracy of the model is investigated using common benchmarks for pseudorandom input sequences. As shown, the average switching activity may be predicted without simulation using either entropy or informational energy averages; the error in prediction is generally small enough to be satisfactory in practice. Future work will be devoted to make the compositional technique more sensitive to different styles of implementing data-path/control hardware.

REFERENCES

- [1] F. N. Najm, 'A Survey of Power Estimation Techniques in VLSI circuits', *IEEE Transactions on VLSI Systems*, vol.2, no.4, pp. 446-455, Dec.1994.
- [2] C.-S. Ding, and M. Pedram, 'A Comparative Study of Switching Activity Estimation Techniques', in *Proc. of PATMOS Conference*, pp.377-396, Germany, Oct.1995

- [3] S. M. Kang, 'Accurate Simulation of Power Dissipation in VLSI Circuits', *IEEE Journal of Solid-State Circuits*, vol.SC-21, no.5, pp. 889-891, 1986.
- [4] A.-C. Deng, Y.-C. Shiau, and H.-H.Loh, 'Time Domain Current Waveform Simulation of CMOS Circuits', in *Proc. IEEE/ACM Intl. Conference on Computer Aided Design*, pp. 208-211, Nov.1988.
- [5] G.Y. Yacoub, and W.H.Ku, 'An Enhanced Technique for Simulating Short-Circuit Power Dissipation', *IEEE Journal of Solid-state Circuits*, Vol. 24, pp. 844-847, June 1989.
- [6] F.N. Najm, 'A Monte Carlo Approach for Power Estimation', *IEEE Transactions on VLSI Systems*, vol.1, no.1, pp. 63-71, Mar. 1993.
- [7] M. Xakellis, and F.N. Najm, 'Statistical Estimation of the Switching Activity in Digital Circuits', in *Proc. ACM/IEEE Design Automation Conference*, pp. 728-733, Nov. 1994
- [8] F.N. Najm, S. Goel, and I.N. Hajj, 'Power Estimation in Sequential Circuits', in *Proc. ACM/IEEE Design Automation Conference*, pp. 635-640, June 1995.
- [9] F.N. Najm, 'Probabilistic Simulation for Reliability Analysis of CMOS Circuits', *IEEE Transactions on CAD*, vol. 9, no. 4, pp. 439-450, Apr. 1990.
- [10] A. Ghosh, S. Devadas, K. Keutzer, and J. White, 'Estimation of Average Switching Activity in Combinational and Sequential Circuits', in *Proc. ACM/IEEE Design Automation Conference*, pp. 270-275, June 1992.
- [11] F. N. Najm, 'Transition Density: A New Measure of Activity in Digital Circuits', *IEEE Transactions on CAD*, vol. 12, no. 2, pp. 310-323, Feb. 1993.
- [12] C.-Y. Tsui, M. Pedram, and A. M. Despain, 'Efficient Estimation of Dynamic Power Dissipation with an Application', in *Proc. IEEE/ACM Intl. Conference on Computer Aided Design*, pp. 224-228, Nov. 1993.
- [13] P. Schneider, U. Schlichtmann, and K. Antreich, 'Decomposition of Boolean Functions for Low Power Based on a New Power Estimation Technique', in *Proc. Intl. Workshop on Low Power Design*, pp. 123-128, April 1994.
- [14] B. Kapoor, 'Improving the Accuracy of Circuit Activity Measurement', in *Proc. ACM/IEEE Design Automation Conference*, pp. 734-739, June 1994.
- [15] T.L. Chou, K. Roy, and S. Prasad, 'Estimation of Circuit Activity Considering Signal Correlations and Simultaneous Switching', in *Proc. IEEE/ACM Intl. Conference on Computer Aided Design*, pp. 300-303, Nov. 1994.
- [16] R. Marculescu, D. Marculescu, and M. Pedram, 'Switching Activity Analysis Considering Spatiotemporal Correlations', in *Proc. IEEE/ACM Intl. Conference on Computer Aided Design.*, pp. 294-299, Nov. 1994.
- [17] R. Marculescu, D. Marculescu, and M. Pedram, 'Efficient Power Estimation for Highly Correlated Input Streams', in *Proc. ACM/IEEE Design Automation Conference*, pp. 628-634, June 1995.
- [18] A. Chandrakasan, et. al, 'HYPER-LP: A System for Power Minimization Using Architectural Transformation', in *Proc. IEEE/ACM Intl. Conference on Computer Aided Design*, pp. 300-303, Nov.1992.
- [19] R. Mehra, J. Rabaey, 'High Level Estimation and Exploration', in *Proc. Intl. Workshop on Low Power Design*, pp.197- 202, April 1994.
- [20] P. Landman, J. Rabaey, 'Power Estimation for High Level Synthesis', in *Proc. European Design Automation Conference*, pp. 361-366, Feb.1993.
- [21] N. Weste and K Eshraghian, 'Principles of CMOS VLSI Design: A Systems Perspective', 2nd Edition, Addison-Wesley Publishing Co., 1994.
- [22] P. Landman and J. Rabaey, 'Black-Box Capacitance Models for Architectural Power Analysis', in *Proc.Intl. Workshop on Low Power Design*, pp. 165-170, April 1994.
- [23] L. Brillouin, 'Science and Information Theory', Academic Press, New York, 1962.
- [24] L. Hellerman, 'A Measure of Computational Work', in *IEEE Trans. on Computers*, vol. C-21, No.5, pp. 439-446, 1972.
- [25] R.W. Cook, M.J. Flynn, 'Logical Network Cost and Entropy', in *IEEE Trans. on Computers*, vol. C-22, No. 9, pp. 823-826, Sept. 1973.
- [26] K.-T. Cheng, V. Agrawal, 'An Entropy Measure for the Complexity of Multi-Output Boolean Functions', in *Proc. ACM/IEEE Design Automation Conference*, pp. 302-305, June 1990.

- [27] V. Agrawal, 'An Information Theoretic Approach to Digital Fault Testing', in *IEEE Trans. on Computers*, vol. C-30, No. 8, pp. 582-587, Aug. 1981.
- [28] K. Thearling, J. Abraham, 'An Easily Computed Functional Level Testability Measure', in *Proc. Intl. Test Conference*, pp. 381-390, 1989.
- [29] A. Papoulis, 'Probability, Random Variables, and Stochastic Processes', McGraw-Hill Co., 1984.
- [30] R. Marculescu, 'A Probabilistic Approach for Power Estimation Under Pseudorandom Input Sequences', Class Project, Fall 1993, Univ. of Southern California.
- [31] O. Onicescu, 'Theorie de l'information. Energie informationelle', Paris, C.R. Acad.Sci., Tome 263, 1966.
- [32] O. Onicescu, V. Stefanescu, 'Elements of Informational Statistics with Applications', in Romanian, Bucharest 1979.
- [33] P. Paulin, and J. Knight, 'Force-directed Scheduling for the Behavioral Synthesis of ASICs', *IEEE Transactions on CAD*, vol. CAD-8, No. 6, pp. 661-679, July 1989.