Power Modeling for High Level Power Estimation†

Subodh Gupta and Farid N. Najm

ECE Dept. and Coordinated Science Lab.
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
sgupta@uivlsi.csl.uiuc.edu, najm@uiuc.edu

Abstract – In this paper, we propose a modeling approach that captures the dependence of the power dissipation of a combinational logic circuit on its input/output signal switching statistics. The resulting power macromodel, consisting of a single four-dimensional table, can be used to estimate the power consumed in the circuit for any given input/output signal statistics. Given a low-level (typically gate-level) description of the circuit, we describe a characterization process by which such a table model can be automatically built. The four dimensions of our table-based model are the average input signal probability, average input transition density, average spatial correlation coefficient and average output zero-delay transition density. This approach has been implemented and models have been built for many benchmark circuits. Over a wide range of input signal statistics, we show that this model gives very good accuracy, with an RMS error of about 4% and average error of about 6%. Except for one out of about 10,000 cases, the largest error observed was under 20%. If one ignores the glitching activity, then the RMS error becomes under 1%, the average error becomes under 5% and the largest error observed in all cases is under 18%.

1. Introduction

With the advent of portable and high-density micro-electronic devices, the power dissipation of very large scale integrated (VLSI) circuits is becoming a critical concern. Modern microprocessors are hot, and their power consumption can exceed 30 or 50 Watts. Due to limited battery life, reliability issues, and packaging/cooling costs, power consumption has become a more critical design concern than speed and area in some applications. Hence to avoid problems associated with excessive power consumption, there is a need for CAD tools to help in estimating the power consumption of VLSI designs.

A number of CAD techniques have been proposed for gate-level power estimation (see [1] for a survey). However, by the time the design has been specified down to the gate level, it may be too late or too expensive to go back and fix high power problems. Hence in order to avoid costly redesign steps, power estimation tools are required that can estimate the power consumption at a high level of abstraction, such as when the circuit is represented only by the Boolean equations. This will provide the designer with more flexibility to explore design trade-offs early in the design process, reducing the design cost and time.

In response to this need, a number of high-level power estimation techniques have been proposed (see [2] for a survey). Two styles of techniques have been proposed, which we refer to as top-down and bottom-up. In the top-down techniques [3, 4], a combinational circuit is specified only as a Boolean function, with no information on the circuit structure, number of gates/nodes, etc.. Top-down methods would be useful when one is designing a logic block that was not previously designed, so that its internal structural details are unknown.

In contrast, bottom-up methods [5–9] are useful when one is reusing a previously-designed logic block, so that all the internal structural details of the circuit are known. In this case, one develops a *power macromodel* for this block which can be used during high-level power estimation (of the overall system in which this block is used), in order to estimate the power dissipation of this block without performing a more expensive gate-level power estimation on it.

The method in [5] uses the power factor approximation technique, which treats all the circuit input bits as digital "white noise" and due to this assumption can give errors of up to 80% in comparison to gate-level tools. Although [6] gives more accurate result, its main disadvantage is that it treats different modules differently, requiring specialized analytical expressions for the power to be provided by the user. Thus, depending upon the functionality of the module, a different type of macromodel (analytical equation) may have to be used.

The method in [7] characterizes the power dissipation of circuits based on input transitions rather

† This work was supported by Rockwell, by Intel Corp., and by the National Science Foundation (MIP 96-23237 & 97-10235).

than input statistics. Since the number of possible input transitions for an n-input combinational circuit is 2^{2n}, they present a clustering algorithm to compress the input transitions into clusters of input transitions that have the same power values (approximately). They use heuristics to implement the clustering algorithm, but it is not clear how efficient the method would be on large circuits.

In [8], the authors present a technique to estimate switching activity and power consumption at the RTL for data path and control circuits, in the presence of glitching activity. To construct a power macromodel, they use both analytical equations and look-up tables. The method is quite good and uses 9 or more variables in the power macromodel. Our independent work has shown that it is possible to construct a look-up table power macromodel with much fewer variables (4 can be enough).

Recently, in [9], the authors presented a macromodel for estimating the cycle-by-cycle power at the RTL. The proposed methodology consists of three steps: module equation form generation and variable selection, variable reduction, and population stratifications. The generated macromodel has 15 variables. They show good accuracy in estimating average and cycle-by-cycle power. The macromodels are dependent on a training vector set, so that the accuracy is compromised if the training set is not similar to the vector set to be applied.

In this paper, we propose a power macromodeling approach that (1) takes into account the effect of the circuit input switching activity and does not treat the circuit inputs as white noise, (2) takes into account input correlation, both spatial and temporal, and (3) is based on a single fixed macromodel template which does not depend on the type of module being analyzed. Our model is table-based. Specifically, we construct a four dimensional look-up table, whose axes are the *average input signal probability* (P_{in}), *average input transition density* (D_{in}),*average input spatial correlation coefficient* (SC_{in}), and *average output zero delay transition density* (D_{out}). For a logic node, the transition density is defined as the average number of logic transitions per unit time [10]. The zero delay transition density refers to the case when the circuit gates are considered to have zero delay, so that only truly required logic transitions (and no hazards or glitches) are observed. From a high-level view, it is reasonable to assume that fast functional simulation will be applied to measure signal switching statistics, so that only the zero delay output density (and not the real delay output density) will be computed. The main advantage of our approach is that all types of circuits are treated in the same way, i.e., we do not use different model equation types for different modules. As a result, the method is very easy to use, and requires no user intervention. Indeed, we will present an automatic characterization procedure by which the macromodel can be built for a given circuit. In this paper we will present an extension of the approach discussed in [11].

The paper is organized as follows. In section 2 we will discuss the macromodeling problem in more detail. In section 3 we will describe the characterization procedure for the models. In section 4 we will evaluate the accuracy of the macromodels and in section 5 we will give some conclusions.

2. POWER MACROMODELING

What should a power macromodel look like? Which features are desirable and which are too expensive and infeasible? To begin with, it is clear that a macromodel should be simple to evaluate, otherwise there would be no advantage in using it and one might as well perform the analysis at the gate level. Furthermore, it must apply over the whole range of possible input signal statistics. Finally, it should consist of a fixed template, in which certain parameter values can be determined by a well-defined and automatic process of *characterization*, without user intervention. We present a macromodel that has all these properties.

2.1 *Power and Input Parameters Relationship*

It is instructive to study the relationship between power and input parameters like average probability and average transition density (see Eq. (17) for definitions) of the primary inputs. Simulations were performed for different values of average input probability and average input density to determine the nature of their relationship with power. Fig. 1 shows the plot of real-delay power dissipation for different values of

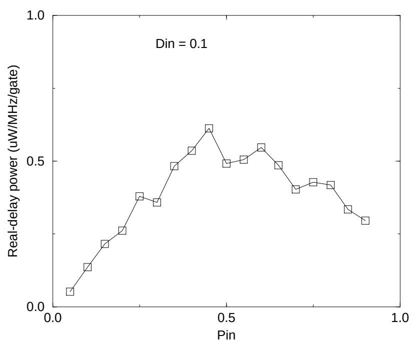

Figure 1. Plot of total power for c6288, for $D_{in} = 0.1$ and different P_{in}.

average input probability and average input density for c6288, a combinational benchmark circuit [12]. Figs. 2 and 3 show the same plot for c3540, another combinational benchmark circuit [12]. It can be seen that the relationship is nonlinear and the plots do not have a consistent shape. Similar results were obtained for other circuits. These results preclude, for instance, the use of a simple linear relationship to relate power to the signal statistics, and led us to consider a table-based approach.

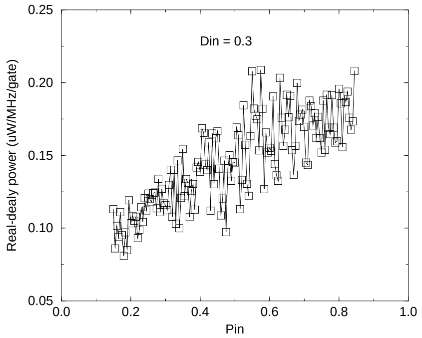

Figure 2. Plot of total power for c3540, for $D_{in} = 0.3$ and different P_{in}.

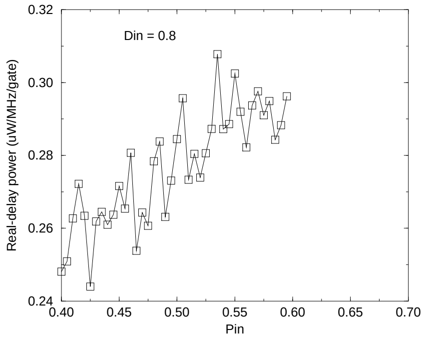

Figure 3. Plot of total power for c3540, for $D_{in} = 0.8$ and different P_{in}.

2.2 *Power Macromodeling Assuming Independence*

Because the power depends on the circuit input switching activity, it is clear that a power macromodel should take the input activity into account. The question is, however, exactly what information about the inputs should be taken into account and included in the macromodel. When the circuit being modeled is small (one or a few gates), then a simple modeling strategy is to create a table that gives the power for every possible input vector pair. In this case, there is no loss of accuracy. However, this strategy cannot be applied to large circuits. A circuit with 32 inputs will have 2^{64} possible input vector pairs, which would be prohibitively expensive to store in a table. This leads to a trade-off between the amount of detail that one includes about the inputs and the accuracy resulting from the model. One possibility is to consider the signal probability $P(x_i)$ and transition density $D(x_i)$ at every input node x_i, and to build a model that depends only on these two variables. Notice that any information about correlations between the input nodes is lost when this is done. Thus, for instance, one could consider building a table which gives the power for every given assignment of input $P(x_i)$ and $D(x_i)$ values. Even in this case, however, such a table-based model would be too expensive, because a circuit with 32 inputs would require a 64-dimensional table.

Given the above observations, we have considered what aggregate compact descriptions of the $P(x_i)$ and $D(x_i)$ values would be sufficient to model the circuit power. For instance, one could consider building a two-dimensional table whose axes would be the average input $P(x_i)$, which we will denote by P_{in}, and the average input $D(x_i)$, to be denoted D_{in}. In this case, two different input assignments of $P(x_i)$ and $D(x_i)$ values, which may lead to different power values, may have the same P_{in} and D_{in} averages, and the table would predict the same power for both assignments, obviously with some error.

We have studied how big this error can be, as follows. Given a gate-level circuit and for a certain fixed P_{in} and D_{in}, we generate a large number (80 or more) of P and D assignments at the circuit inputs that each have averages equal to the specified P_{in} and D_{in}. We then perform an accurate power estimation for each assignment using a Monte Carlo gate-level (with full delay model) simulation technique [13]. The average of the resulting power values is a good candidate value to store in the table. For each of the estimated power values, any deviation from this average value is considered to be an "error" relative to this table. The root-mean-square (RMS) and maximum errors for ISCAS85 circuits [12] (see Table 1 for details of these circuits) are reported in Table 2, for $P_{in} = 0.4$ and $D_{in} = 0.4$. A density of 0.4 means that the node makes an average of 4 transitions in 10 consecutive clock cycles. The largest RMS error is about 17% and the largest maximum error is -40%.

Table 1. Details of the ISCAS85 circuits.

Circuit	Function	#inputs	#outputs	#gates
c432	Interrupt control	36	7	160
c880	ALU	60	26	383
c1908	Error correction	33	25	880
c2670	ALU and control	233	140	1193
c3540	ALU	50	22	1669
c5315	ALU	178	123	2307
c6288	Multiplication	32	32	2406
c7552	ALU	207	108	3512
c499	Error detection	41	32	202
c1355	Error detection	41	32	546

Table 2. RMS and maximum error in the 2-d table approach, when total power is estimated.

Circuit	P_{in}	D_{in}	RMS.Error	Max.Error
c432	0.4	0.4	1.61%	34.88%
c880	0.4	0.4	1.77%	40.46%
c1908	0.4	0.4	1.74%	16.80%
c2670	0.4	0.4	2.43%	-31.61%
c3540	0.4	0.4	2.96%	35.77%
c5315	0.4	0.4	1.76%	20.94%
c6288	0.4	0.4	16.6%	-40.04%
c7552	0.4	0.4	3.37%	19.02%

The power estimator (simulator) used to generate this table uses a scalable-delay timing model that depends on fanout and gate output capacitance. Thus, it captures the glitching power accurately (multiple transitions per cycle due to unequal delay from the inputs to an internal node). The glitching power is hard to account for in a high-level model. This is why such a high RMS error is seen for c6288, in which some internal nodes make up to 20 transitions per cycle. The errors improve considerably if the power estimates are based on a zero-delay timing model, in which the glitches are excluded, as shown in Table 3. The largest RMS error is now 1% and the largest maximum error is 27%.

Table 3. RMS and maximum error in the 2-d table approach, when zero-delay power is estimated.

Circuit	P_{in}	D_{in}	RMS.Error	Max.Error
c432	0.4	0.4	0.59%	16.02%
c880	0.4	0.4	0.85%	27.5%
c1908	0.4	0.4	0.46%	-7.28%
c2670	0.4	0.4	0.92%	-18.82%
c3540	0.4	0.4	0.83%	-19.07%
c5315	0.4	0.4	0.47%	10.88%
c6288	0.4	0.4	0.72%	-16.82%
c7552	0.4	0.4	1.01%	-15.54%

In any case, with such a high RMS error in the general delay case, the total power estimation using Table 2 is too inaccurate. The simple 2-dimensional table approach is too simplistic. Another parameter is needed by which we can accurately model the variation of the power due to various input P and D assignments. We have found that if one more dimension is added to the table, reasonably good accuracy can be obtained. The third axis is the average output transition density over all the circuit output nodes, measured from a zero-delay (functional) simulation of the circuit, and which we will denote by D_{out}. The stipulation that D_{out} corresponds to zero-delay is not optional, but rather required for the following reason. We envision that during high-level, say RTL, power estimation, one would perform an initial step of estimating the signal statistics at the visible RTL nodes from a high-level functional simulation. These (zero-delay) statistics would then be applied to the power macromodel in order to estimate the power. Thus, the power model will be given by:

$$P_{avg} = f(P_{in}, D_{in}, D_{out}) \tag{1}$$

In order to study the accuracy in this 3-d approach, and to perform a direct comparison with Tables 2 and 3, we will show the errors in the estimation for the same $P_{in} = 0.4$ and $D_{in} = 0.4$ specifications as before. The value of D_{out} will naturally be different in different runs. For each circuit, we selected the largest subset of cases that has the same (approximately) D_{out} value and examined the errors based on the results in that subset. It is clear from Table 4 that the errors are much less now, and the RMS error in c6288 is now reduced to an acceptable 6%. For comparison with Table 3, the errors in the zero-delay power are given in Table 5. The RMS error is now below 0.77% and the maximum error is under about 12%.

Table 4. RMS and maximum error in the 3-d table approach, when total power is estimated.

Circuit	P_{in}	D_{in}	D_{out}	RMS.Error	Max.Error
c432	0.4	0.4	0.44	0.97%	16.48%
c880	0.4	0.4	0.32	1.58%	27.87%
c1908	0.4	0.4	0.44	1.18%	12.71%
c2670	0.4	0.4	0.37	1.78%	-18.82%
c3540	0.4	0.4	0.44	1.94%	-20.33%
c5315	0.4	0.4	0.42	1.76%	17.16%
c6288	0.4	0.4	0.44	6.05%	-33.54%
c7552	0.4	0.4	0.42	2.97%	-15.67%

Table 5. RMS and maximum error in the 3-d table approach, when zero-delay power is estimated.

Circuit	P_{in}	D_{in}	D_{out}	RMS.Error	Max.Error
c432	0.4	0.4	0.44	0.33%	4.90%
c880	0.4	0.4	0.32	0.55%	9.87%
c1908	0.4	0.4	0.44	0.19%	-3.23%
c2670	0.4	0.4	0.37	0.65%	-9.70%
c3540	0.4	0.4	0.44	0.47%	-12.37%
c5315	0.4	0.4	0.42	0.45%	6.32%
c6288	0.4	0.4	0.44	0.45%	-10.18%
c7552	0.4	0.4	0.42	0.77%	-8.82%

2.3 *Power Macromodeling For Correlated Inputs*

In the previous section we assumed that the primary inputs are independent, but in practice the primary inputs can be correlated. For example, the primary inputs could be the output of another circuit block, which can be very highly correlated. Fig. 4 compares the correlated and 3-d table-based power values for all ISCAS-85 circuits, over a wide range of P_{in}, D_{in}, and D_{out} values. An enlarged view of the lower section of the Fig. 4 is shown in Fig. 5. It can be seen from the figures that the 3-dimensional table-based macromodel gives erroneous estimate of the power when primary inputs are correlated. Table 6 gives the RMS, average and maximum error, when the inputs are correlated and the total power is estimated using the 3-d table-based macromodel, over a wide range of P_{in}, D_{in}, and D_{out} values. It can be seen from the table that the error is quite high. This led us to consider other parameters to be included in the macromodel.

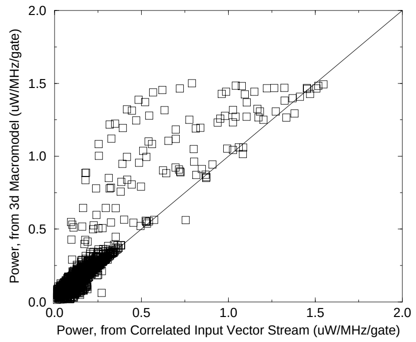

Figure 4. Power comparison between correlated input vector stream and 3-d macromodel, when total power is estimated.

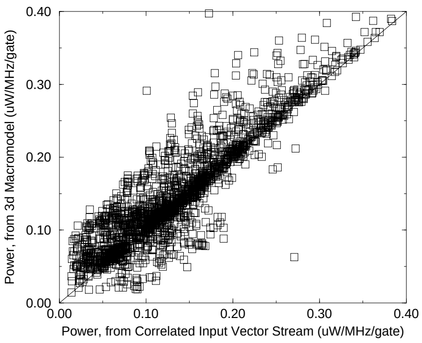

Figure 5. Power comparison between correlated input vector stream and 3-d macromodel, when total power is estimated.

Table 6. RMS, average and maximum error when total power of correlated input vector stream is estimated using 3-d macromodel.

Circuit	RMS.Error	Average Error	Max.Error
c432	3.84%	35.5%	122.16%
c880	2.00%	16.26%	73.9%
c1908	3.73%	25.75%	114.78%
c2670	4.46%	27.08%	116.44%
c3540	2.936%	20.59%	120.01%
c5315	3.72%	21.72%	121.75%
c6288	41.4%	90.17%	226.64%
c7552	4.56%	28.73%	124.34%
c499	3.36%	43.15%	160.79%
c1355	2.846%	29.66%	134.71%

The primary inputs can be either temporally or spatially correlated. A signal x is said to be *temporally correlated* if an event (occurrence of certain logic state) at a given time is correlated to an event at some past time and is said to be *spatially correlated* to another signal y if their events are correlated.

2.3.1 *Temporal Correlation*

In the case of temporal correlation, we will consider only correlations across one clock edge. For temporally correlated primary inputs, define TC_i for the ith input, as:

$$TC_i = \mathcal{P}\left\{x_i^t \wedge x_i^{t-1} = 1\right\} \quad (2)$$

where $t-1$ and t are consecutive clock cycles and where $\mathcal{P}\{\cdot\}$ denotes probability. Temporal correlation coefficient (γ_i) for ith input is defined as [14]:

$$\gamma_i = \frac{\mathcal{P}\left\{x_i^t \wedge x_i^{t-1} = 1\right\} - P(x_i)^2}{P(x_i)(1 - P(x_i))} \quad (3)$$

In (3), $P(x_i)$ is the probability at an input node x_i, which is known, as individual input probabilities are required to determine P_{in} for the 3-dimensional table based power macromodel and the only quantity which is unknown is $\mathcal{P}\left\{x_i^t \wedge x_i^{t-1} = 1\right\}$. Therefore, γ_i can be estimated accurately, if we can determine TC_i. But, we will show now that TC_i can be uniquely determined from the knowledge of $P(x_i)$ and $D(x_i)$.

Proposition 1. *For any primary input node:*

$$TC_i = P(x_i) - \frac{D(x_i)}{2} \quad (4)$$

where TC_i, $P(x_i)$ and $D(x_i)$ are the temporal correlation, signal probability and transition density, respectively.

Proof: Let us denote the probability of a low-to-high transition by P_{lh}, and the probability of a high-to-low transition by P_{hl}. Since a low-to-high transition is eventually followed by a high-to-low transition, then:

$$P_{lh} = P_{hl} \quad (5)$$

The transition density can be expressed as:

$$D(x_i) = P_{lh} + P_{hl} = 2P_{lh} = 2\left[P(x_i) - TC_i\right] \quad (6)$$

$$\Rightarrow TC_i = P(x_i) - \frac{D(x_i)}{2} \quad (7)$$

Hence proved.■

Therefore, temporal correlation at the primary inputs is taken care by $P(x_i)$ and $D(x_i)$ and we do not need an additional parameter to represent it.

2.3.2 *Spatial Correlation*

We will consider only pairwise correlations. We define SC_{ij}, the spatial correlation between the ith and jth inputs as:

$$SC_{ij} = \mathcal{P}\left\{x_i \wedge x_j = 1\right\}, \tag{8}$$

i.e., the probability of both inputs being high simultaneously.

The reason for considering SC_{ij} as the measure of spatial correlation coefficient follows from the definition of correlation coefficient [14]:

$$\rho_{ij} = \frac{\mathcal{P}\left\{x_i \wedge x_j = 1\right\} - P(x_i)P(x_j)}{\sqrt{P(x_i)P(x_j)\left(1 - P(x_i)\right)\left(1 - P(x_j)\right)}} \tag{9}$$

From the definition given in (8), it is clear that SC_{ij} is sufficient to capture ρ_{ij}.

As the number of primary inputs increases, the number of SC_{ij} parameters will increase quadratically. We have found empirically that if we consider SC_{in} (*average spatial correlation coefficient*, i.e, average of all SC_{ij} terms), as the fourth parameter in the power macromodel, sufficient accuracy can be obtained for estimating the power of highly correlated primary inputs. Thus, our table-based power macromodel in presence of the fourth parameter looks as follows:

$$P_{avg} = f(P_{in}, D_{in}, SC_{in}, D_{out}) \tag{10}$$

3. CHARACTERIZATION

We assume that the combinational circuit is embedded in a larger sequential circuit, so that its input nodes are the outputs of latches or flip-flops and that they make at most one transition per clock cycle. We assume that the sequential design is a single clock system and ignore clock skew, so that the combinational circuit inputs $x_1, x_2, \ldots, x_n$ switch only at time 0.

At this point it is helpful to recall some definitions. The signal probability $P(x_i)$ at an input node x_i is defined as the average fraction of clock cycles in which the final value of x_i is a logic high. The transition density $D(x_i)$ at an input node x_i is defined as the average fraction of cycles in which the node makes a logic transition (its final value is different from its initial value). For brevity, in this section we will write P_i and D_i to represent $P(x_i)$ and $D(x_i)$. Both P_i and D_i are real numbers between 0 and 1.

Because the input signals x_i make at most a single transition per cycle, there is a special relationship between probability and density, given by:

$$\frac{D_i}{2} \leq P_i \leq 1 - \frac{D_i}{2} \tag{11}$$

The derivation of this property is rather simple, as follows. Let μ_1 (μ_0) be the average number of consecutive clock cycles that an input node remains high (low). Through a minor extension of the results in [10] to the case of discrete time signals, $P(x)$ and $D(x)$ at input node x are given by:

$$P(x) = \frac{\mu_1}{\mu_0 + \mu_1} \tag{12}$$

$$D(x) = \frac{2}{\mu_0 + \mu_1} \tag{13}$$

from which it follows that:

$$\mu_1 = \frac{2P(x)}{D(x)} \tag{14}$$

$$\mu_0 = \frac{2\left(1 - P(x)\right)}{D(x)} \tag{15}$$

Since $\mu_1 \geq 1$ and $\mu_0 \geq 1$, (14) and (15) lead to the required result (11).

One can rewrite (11) as:

$$D_i \leq 1 - 2\left|P_i - 0.5\right| \tag{16}$$

so that for a given $P(x)$, $D(x)$ is restricted to the shaded region shown in Fig. 6.

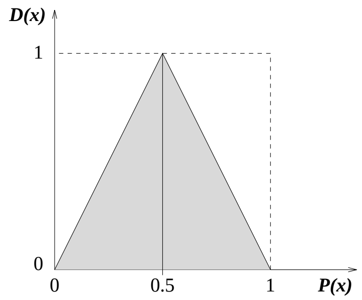

Figure 6. Relationship between density and probability for discrete-time signals.

We also recall the definitions of the average input probability, denoted P_{in}, and average input density, denoted D_{in}, as follows:

$$P_{in} = \frac{1}{n}\sum_{i=1}^{n} P_i \qquad D_{in} = \frac{1}{n}\sum_{i=1}^{n} D_i \tag{17}$$

where n is the number of input nodes. It is clear from (11) that similar bounds hold for P_{in} and D_{in}:

$$\frac{D_{in}}{2} \leq P_{in} \leq 1 - \frac{D_{in}}{2} \tag{18}$$

from which we also have:

$$D_{in} \leq 1 - 2\left|P_{in} - 0.5\right| \tag{19}$$

Similarly we can derive a special relationship between SC_{in} and P_{in}, i.e., given P_{in} we can find lower and upper bounds for SC_{in}. Because SC_{in} is a probability it can take values only between 0 and 1. Before describing the bounds, we first recall the definition of SC_{in}:

$$SC_{in} = \frac{2}{n\left(n-1\right)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}\mathcal{P}\left\{x_i = 1, x_j = 1\right\} \tag{20}$$

where n is the number of primary inputs.

Let us consider that we have to generate a block of N consecutive input vectors, with each vector consisting of 1s and 0s, and let us denote the kth vector by V_k. SC_{in} can be written in terms of the input vectors as:

$$SC_{in} = \lim_{N\to\infty} SC_{in}^{N} \tag{21}$$

where:

$$\begin{aligned} SC_{in}^{N} &= \frac{1}{N}\sum_{k=1}^{N}\frac{2}{n\left(n-1\right)}\sum_{i=1}^{n}\sum_{j=i+1}^{n} x_{i,k}x_{j,k} \\ &= \frac{2}{n\left(n-1\right)N}\sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{j=i+1}^{n} x_{i,k}x_{j,k} \end{aligned} \tag{22}$$

and where $x_{i,k}$ is the ith bit in the kth vector. Notice that $\sum_{i=1}^{n}\sum_{j=i+1}^{n} x_{i,k}x_{j,k}$ = number of bit pairs, in kth vector, that are $(1,1)$. Therefore,

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n} x_{i,k}x_{j,k} = \frac{n_1\left(k\right)\left(n_1\left(k\right)-1\right)}{2} \tag{23}$$

where $n_1\left(k\right)$ = number of $1s$ in V_k.

By substituting (23) into (22), we get:

$$SC_{in}^{N} = \frac{2}{Nn\left(n-1\right)}\sum_{k=1}^{N}\frac{n_1\left(k\right)\left(n_1\left(k\right)-1\right)}{2} \tag{24}$$

At this point, it will be helpful to define P_{in}^{N}. For a block of N vectors, P_{in} can be written as:

$$P_{in} = \lim_{N\to\infty} P_{in}^{N} \tag{25}$$

where:

$$P_{in}^{N} = \frac{1}{N}\sum_{k=1}^{N}\frac{n_1\left(k\right)}{n} \tag{26}$$

Notice that, for large N:

$$\frac{1}{N}\sum_{k=1}^{N} n_1\left(k\right) \approx nP_{in} \tag{27}$$

It can be shown from (24) & (26) that, if we allow $n_1(k)$ to take real non-integer values, then the minimum value of SC_{in}^{N} occurs when, for all k (see Appendix A for proof):

$$n_1\left(k\right) = nP_{in}^{N} \tag{28}$$

Therefore, a lower bound on SC_{in}^{N} is given by:

$$SC_{in}^{N} \geq \frac{nP_{in}^{N}\left(nP_{in}^{N}-1\right)}{n\left(n-1\right)} \tag{29}$$

For large values of N, this leads to:

$$SC_{in} \geq \frac{nP_{in}^{2} - P_{in}}{\left(n-1\right)} \tag{30}$$

To compute an upper bound on SC_{in}^{N}, we start with the observation that the maximum value of SC_{in}^{N} in (24) will occur when as many $n_1(k)$s as possible are set to their maximum value n, because of the quadratic term. Since not all $n_1(k)$s can be set to n due to (26), the largest SC_{in}^{N} is achieved by having $m < N$ vectors have $n_1(k) = n$ 1s, one vector contain the remaining $r < n$ 1s, and the remaining vectors contain all 0s. In other words, m is the largest integer for which $mn + r = \sum_{k=1}^{N} n_1(k) = NnP_{in}^{N}$, where $0 < r < n$ is an integer. With this, $m = \lfloor NP_{in}^{N} \rfloor$, and the largest possible value of SC_{in}^{N} is given by:

$$SC_{in}^{N} \leq \frac{mn\left(n-1\right) + r\left(r-1\right)}{Nn\left(n-1\right)} = \frac{m}{N} + \frac{r(r-1)}{Nn(n-1)} \tag{31}$$

From this, it follows that:

$$SC_{in} = \lim_{N\to\infty} SC_{in}^{N} \leq P_{in} \tag{32}$$

due to the fact that $m/N = P_{in}^{N} - (r/Nn)$ so that $\lim_{N\to\infty} m/N = P_{in}$.

Combining the lower and upper bounds gives:

$$\frac{nP_{in}^{2} - P_{in}}{\left(n-1\right)} \leq SC_{in} \leq P_{in} \tag{33}$$

The shaded region in Fig. 7 shows the feasible region for P_{in} and SC_{in}. Shown in Fig. 8 is the three-dimensional plot showing the relationship between P_{in}, D_{in}, and SC_{in}. The two shaded surfaces are the lower and upper bounds for SC_{in} for different values of P_{in} and D_{in}. It is evident from the figure that D_{in}

does not have any effect on SC_{in}. The surface in the $(P_{in}\ D_{in})$ plane shows the relationship between P_{in} and D_{in} as given by (18).

Thus, the 4-dimensional table with axes P_{in}, D_{in}, SC_{in} and D_{out} will not be completely full, and the choices of P_{in}, D_{in}, and SC_{in} during characterization will have to satisfy the above constraints (18) and (33). We subdivide the probability, density and spatial correlation axes between 0 and 1 into intervals of size 0.1, so that we form a $10 \times 10 \times 10$ grid in the $(P_{in}, D_{in}, SC_{in})$ plane. This choice is rather an arbitrary one, which we have found works well. Only a fraction of these points are valid, namely those that fall inside the shaded regions in Figs. 7 and 8. Each valid grid point will correspond to a column of cells in the table along the D_{out} axis as shown in Fig. 9.

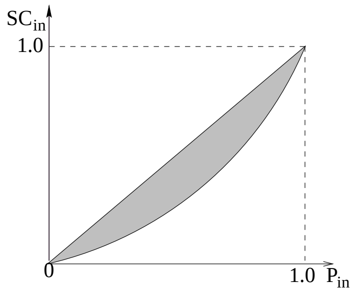

Figure 7. Relationship between probability and spatial correlation for discrete-time signals.

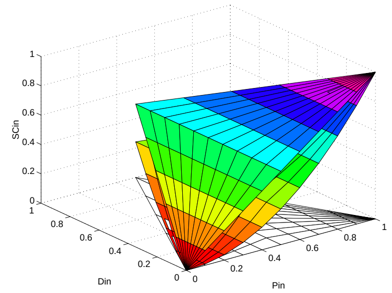

Figure 8. Relationship between probability, density and spatial correlation for discrete-time signals.

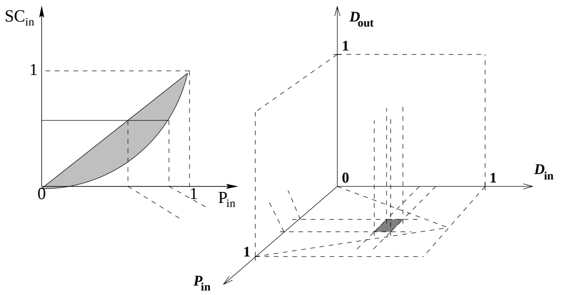

Figure 9. Four dimensional power macromodel.

For each valid grid point in the $(P_{in}, D_{in}, SC_{in})$ space, we generate blocks of input vectors such that the average probability, density and spatial correlation at the primary inputs are equal to P_{in}, D_{in}, and SC_{in} respectively. Using these vectors, the circuit power is computed using Monte Carlo power estimation [13], and the value of D_{out} is computed as the average of the individual (zero-delay) density values at the circuit outputs, also found during the Monte Carlo analysis. The value of D_{out} is rounded to the nearest grid point on the D_{out} axis, and the power value obtained is associated with the resulting cell location $(P_{in}, D_{in}, SC_{in}, D_{out})$ in the table. Eventually, a number of power values may be associated with a single cell in the table. At the end of the characterization, every cell is filled with the average of the power values associated with it. Some cells may have no power values associated with them, in which case their contents are left at zero. When it comes time to use the table, interpolation and extrapolation can be used to find the power for a $(P_{in}, D_{in}, SC_{in}, D_{out})$ combination which does not exist in the table. In the next section, we will show a number of results that demonstrate the accuracy of this approach over a wide range of input statistics, in which interpolation and extrapolation were used whenever required.

The above characterization process is straightforward, except for the generation of the block of input vectors at the primary inputs such that the average values of probability, density, and spatial correlation are equal to P_{in}, D_{in}, and SC_{in} respectively.

Mathematically, the problem can be stated as to generate a block of N input vectors (as shown in Fig. 10) such that they satisfy the following requirements:

$$\begin{aligned} P_{in}^N &\approx P_{in} \\ D_{in}^N &\approx D_{in} \\ SC_{in}^N &\approx SC_{in} \end{aligned} \tag{34}$$

where P_{in}, D_{in}, and SC_{in} are the required average signal probability, average transition density and average spatial correlation coefficient, respectively, at the primary inputs which satisfy (18) and (33). Similarly, P_{in}^N, D_{in}^N, and SC_{in}^N are the averages obtained from the generated input vectors.

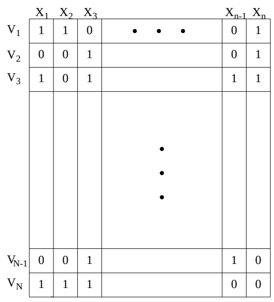

Figure 10. A block of N input vectors.

We have developed a heuristic technique to generate blocks of input vectors satisfying (34). Fig. 11 shows a histogram of the Euclidean distance between $(P_{in}, D_{in}, SC_{in})$ and $(P_{in}^N, D_{in}^N, SC_{in}^N)$, for blocks of input vector of size $N = 100$, over a wide range of P_{in}, D_{in} and SC_{in} values. It is clear from the figure that for most cases the distance is near zero, and that the maximum error is under 5%, thus demonstrating the accuracy of this technique. For more details on this, refer to [15].

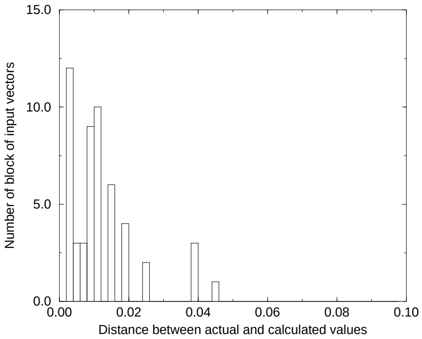

Figure 11. Distance distribution between $(P_{in}, D_{in}, SC_{in})$ and $(P_{in}^N, D_{in}^N, SC_{in}^N)$.

Table 7 gives the execution times for the ISCAS-85 circuits, under the column named 'Time', for building the look-up table based macromodel. The execution times are on a SUN Ultra Sparc 1 with 64 MB of RAM. It can be seen from the table that the largest execution times are required for c7552 and c6288 respectively. This is due to the fact that it is very time consuming to compute the power dissipation for these circuits. The number of iterations (power estimation runs) required to build the macromodel is the same for all the circuits, including c7552 and c6288. If one uses a more efficient power estimator, the overall time to build the macromodel would be reduced. In any case, it should be kept in mind that the time required to build the macromodel is a one-time up-front cost.

4. Model Accuracy Evaluation

In this section, we report the results of the 4-dimensional power macromodeling approach on the ISCAS-85 circuits. We have implemented this approach and built the power macromodels (4-dimensional look-up tables) for a number of combinational circuits. In order to study the accuracy over a wide range of signal statistics, we randomly generated blocks of input vectors at the circuit inputs while covering a wide range of P_{in}, D_{in}, and SC_{in} values that satisfy (18) and (33). Approximately 1000 such valid blocks of input vectors were generated this way for every ISCAS-85 circuits, for which the power was estimated from gate-level Monte Carlo simulation; the Monte Carlo simulation also provides accurate estimation of D_{out}. The power values predicted by the look-up table were compared to those from simulation, and the RMS, absolute average and maximum errors were computed.

The results are summarized in Table 7 for the case when total power is estimated. It is seen that the RMS error is very good, under about 5%. The largest maximum error is at 22.56% for c432, because the estimated power value is very small and a slight difference in power value causes a lot of error. The average error in all cases is less than 6%, which shows the accuracy of our macromodeling approach. The combined scatter plot of all ISCAS-85 circuits showing the accuracy of this approach is shown in Fig. 12. An enlarged view of the lower section of this plot is given in Fig. 13. Both these plots report normalized power values, so that the results for all the circuits can be examined on the same plot.

Table 7. Accuracy of the 4-d look-up tables, when total power is estimated.

Circuit	RMS.Error	Average Error	Max.Error	Time
c432	0.868%	5.56%	22.56%	11.75hrs
c880	0.647%	3.73%	14.64%	6.24hrs
c1908	0.729%	3.85%	16.89%	4.59hrs
c2670	0.738%	3.08%	11.52%	14.23hrs
c3540	0.802%	3.61%	16.53%	21.32hrs
c5315	0.612%	2.48%	-14.58%	16.21hrs
c6288	4.14%	3.75%	18.23%	34.4hrs
c7552	0.847%	3.03%	-16.58%	58.4hrs
c499	0.497%	4.05%	16.4%	2.3hrs
c1355	0.5167%	4.19%	15.6%	2.09hrs

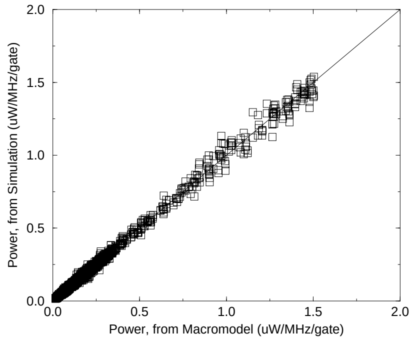

Figure 12. Agreement between the 4-d table and accurate power estimation, when total power is estimated.

For completeness, the accuracy of the macromodels when zero-delay power is estimated is shown in Table 8 and in the scatter plot in Fig. 14. Over a wide range of signal statistics, the RMS error is below 0.60%, the average error is under 5% and the maximum error is under 18%. The scatter plot also shows excellent agreement.

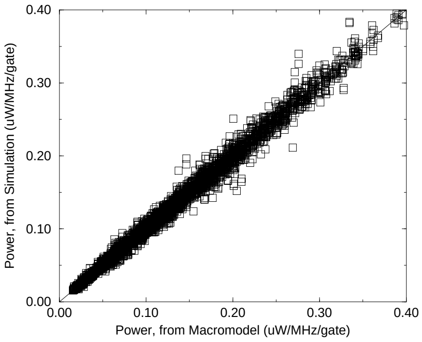

Figure 13. Agreement between the 4-d table and accurate power estimation, when total power is estimated.

Table 8. Accuracy of the 4-d look-up tables, when zero-delay power is estimated.

Circuit	RMS.Error	Average Error	Max.Error
c432	0.428%	4.409%	17.35%
c880	0.519%	3.62%	13.97%
c1908	0.461%	3.73%	15.69%
c2670	0.307%	2.18%	10.16%
c3540	0.413%	3.22%	15.55%
c5315	0.29%	2.08%	-12.20%
c6288	0.332%	2.218%	17.37%
c7552	0.23%	2.65%	-14.32%
c499	0.45%	3.95%	16.34%
c1355	0.383%	4.03%	15.04%

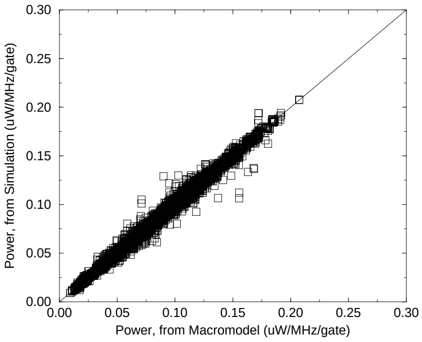

Figure 14. Agreement between the 4-d table and accurate power estimation, when zero-delay power is estimated.

5. CONCLUSION

Since gate-level power estimation can be time-consuming and because power estimation from a high level of abstraction is desirable so as to reduce design time and cost, we have proposed a power macromodeling approach for combinational circuits with synchronous inputs. Our macromodel consists of a 4-dimensional look-up table with axes for average input signal probability, average input transition density, average input spatial correlation coefficient and average output (zero-delay) transition density. A novel and significant aspect of this approach is that we use the same model template for all types of combinational circuits, and no specialized analytical expressions are required. Another important fact is that this model works for all possible signal switching statistics.

We have shown why it is advantageous to use a 4-d table, and described an automatic procedure for building the 4-d macromodel, without the need for user intervention. Once the model for a combinational block has been built, it can be used to estimate power during high-level power estimation, based on signal statistics that are computed from a high-level functional simulation. Over a wide range of input/output signal statistics, we have shown that this model gives very good accuracy, with an RMS error of about 4%. Except for one out of about 10,000 cases, the largest error observed was under 20%. The average error was under 6%. If one ignores the glitching activity, then the RMS error becomes under 0.60%, the average error under 5% and the largest maximum error under 18%.

APPENDIX A

We will derive the values of $n_1(k)$, for which SC_{in}^N and hence SC_{in} takes its minimum value, in support of the result (28). We start by writing (24) as:

$$\sum_{k=1}^{N} n_1^2(k) - \sum_{k=1}^{N} n_1(k) = n(n-1)\,NSC_{in}^N \qquad (A.1)$$

From (26) we have:

$$\sum_{k=1}^{N} n_1(k) = nNP_{in}^N \qquad (A.2)$$

which is a constant. Therefore, the minimization problem becomes:

$$\begin{aligned} &minimize \qquad \sum_{k=1}^{N} n_1^2(k) \\ &s.t. \qquad \sum_{k=1}^{N} n_1(k) = nNP_{in}^N \end{aligned} \qquad (A.3)$$

Proposition A1. *If $n_1(k)$ are allowed to take real non-integer values, then the minimum value of $\sum_{k=1}^{N} n_1^2(k)$, subject to (A.2), occurs when for all k:*

$$n_1(k) = nP_{in}^N \qquad (A.4)$$

Proof: The problem given by (A.3) is a constrained minimization problem. Because it is a convex problem it can be solved by converting it (by introducing a Lagrangian) into an unconstrained problem [16], leading to:

$$\textit{minimize} \qquad \sum_{k=1}^{N} n_1^2\left(k\right) - \lambda \left(nNP_{in}^{N} - \sum_{k=1}^{N} n_1\left(k\right) \right) \tag{A.5}$$

where λ is a constant. Differentiating $\sum_{k=1}^{N} n_1^2\left(k\right) - \lambda \left(nNP_{in}^{N} - \sum_{k=1}^{N} n_1\left(k\right) \right)$ with respect to $n_1\left(k\right)$ and setting it equal to 0 we get:

$$n_1\left(k\right) = -\frac{\lambda}{2} \tag{A.6}$$

Plugging this value of $n_1\left(k\right)$ in (A.2), we get

$$\lambda = -2nP_{in}^{N} \tag{A.7}$$

$$\Rightarrow n_1\left(k\right) = nP_{in}^{N} \tag{A.8}$$

Hence proved. ■

REFERENCES

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

[2] P. Landman, "High-level power estimation," *International Symposium on Low Power Electronics and Design*, pp. 29–35, Monterey, CA, August 12–14, 1996.

[3] M. Nemani and F. Najm, "Towards a High-Level Power Estimation Capability," *IEEE Transactions on CAD* , vol. 15 pp. 588-598, June 1996.

[4] D. Marculescu, R. Marculescu and M. Pedram, "Information Theoretic Measures of Energy Consumption at Register Transfer Level," *ACM/IEEE International Symposium on Low Power Design*, pp. 87-92, April 1995.

[5] S. R. Powell and P. M. Chau, "Estimating Power Dissipation of VLSI signal Processing Chips: The PFA technique," *VLSI Signal Processing IV*, pp. 250-259, 1990.

[6] P. E. Landman and J. M. Rabaey, "Architectural Power Analysis: The Dual Bit Type Method," *IEEE Transactions on VLSI*, vol. 3 pp. 173-187 June 1995.

[7] H. Mehta, R. M. Owens and M. J. Irwin, "Energy Characterization based on Clustering," *33rd ACM/IEEE Design Automation Conference*, pp. 702-707, June 1996.

[8] A. Raghunathan, S. Dey and N. K. Jha, "Register-Transfer Level Estimation Techniques for Switching Activity and Power Consumption," *IEEE International Conference on Computer-Aided Design*, pp. 158-165, November 1996.

[9] Q. Qiu, Q. Wu, Chih-S. Ding, and M. Pedram, "Cycle-accurate macro-models for RT-level power analysis," *Proc. International Symposium on Low Power Electronics and Design*, pp. 125–130, 1997.

[10] F. Najm, "Transition Density: A New Measure of Activity in Digital Circuits," *IEEE Trans. on CAD*, vol. 12, pp. 310-323, Feb. 1993.

[11] S. Gupta and F. Najm, "Power Macromodeling for High Level Power Estimation," *34th ACM/IEEE Design Automation Conference*, pp. 365-370, June 1997.

[12] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran," *IEEE International Symposium on Circuits and Systems*, pp. 695-698, June 1985.

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

[14] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd edition. New York: McGraw-Hill, 1991.

[15] S. Gupta and F. Najm, "Power Macromodeling for High Level Power Estimation," University of Illinois, Coordinated Science Laboratory, Report #UILU-ENG-97-2229, September 1997.

[16] D. G. Luenberger, *Linear and nonlinear programming*, 2nd edition. Reading, MA: Addison-Wesley, 1984.