Regression models for behavioral power estimation

L. Benini, A. Bogliolo, M. Favalli* and G. De Micheli

CSL - Stanford University - Stanford CA 94305-9030 USA

* DEIS - University of Bologna - Viale Risorgimento 2, 40136 Bologna, Italy

Abstract

Behavioral power estimation is required to help the designer in making important architectural choices. In this work we propose an accurate and general behavioral power modeling approach especially suited for synthesis-based design flows making use of a library of hard macros implementing behavioral operators. Power dissipation models are pre-characterized and back-annotated in a preliminary step. Accurate information on the power dissipation of the used macros can then be collected during behavioral simulation of the synthesized circuit. Our characterization and modeling methodology is based on the theory of linear regression. Optimal linear power models are obtained with methods of least squares fitting and its generalization to a recursive procedure called *tree regression*. The behavioral power models are available within PPP, a multilevel simulation engine for power estimation fully compatible with Verilog XL.

1 Introduction

A critical feature for the success of behavioral synthesis tools is the capability of early estimating the power dissipation of large digital systems. In this work we present a novel approach to behavioral power modeling especially suited for synthesis-based design methodologies.

In the design of large digital systems, building blocks are typically described by behavioral models. For instance, at the RTL level the circuit behavior is described by means of arithmetic operators and registers controlled by loop and conditional structures. RTL models are cycle-accurate and enable behavioral simulation orders of magnitude faster than gate-level simulation. Since fully-functional RTL models are available generally much earlier than their gate-level counterparts, obtaining power data during RTL simulation is an attractive possibility.

Techniques for power estimation based on behavioral models have been recently proposed. While in earlier approaches [1] the effect of input signals statistics was not taken into account, Landman et al. [2] proposed a technique that accounts for signal statistics and showed that power is strongly dependent on such information. Unfortunately, the applicability of this approach depends on a set of assumptions on data representation and signal statistics, and relies on human knowledge for the formulation of basic behavioral models that are subsequently automatically optimized.

Our modeling technique allows accurate power estimation in systems where the data representation and signal statistics do not satisfy the requirements for the applicability of the methods proposed in [2]. We start from a library of *hard macros* implementing the behavioral operators. Our characterization procedure is run once for all on the library elements (for which we assume the availability of a gate or circuit level representation). Power models are then extracted and backannotated in the behavioral representations of the library elements. The backannotated models can be run within RTL simulation and provide a high level power estimate. Notice that characterization of hard macros can be performed once for all by the library vendor. This is not the case for soft-macros, that are generated from synthesizable HDL at design time. In this work we do not deal with soft macros.

Our approach is a generalization of well-known linear regression techniques. We abstract all information on the internal structure of the unit (*i.e.*, we assume that the circuit is a *black box*). As a consequence no human knowledge is required and the model extraction procedure is fully automatic. For a class of circuits the accuracy of the regression model can be improved if different regression equations are obtained for different modes of operations. We define a new characterization procedure called *tree regression* that captures this kind of behavior.

The experimental results show that behavioral power estimation is a feasible alternative to gate-level (or circuit level) techniques even in cases where no preliminary assumptions on data representation and signal statistics can be exploited in the precharacterization phase. Although the loss of accuracy is sizable, our models always perform better than simple estimates on average power.

We have embedded the behavioral power estimation tool in PPP [5], a multilevel power estimation engine designed to assist the designer with accurate power information during the complete design process, from the specification to the final gate-level implementation.

2 Previous work

In the simplest kind of RTL models [1, 3], the power dissipation of a functional unit is approximated with a single fitting coefficient P, namely, the average power dissipation. The value of P is generally computed by simulating the unit with a long sequence of random input patterns possibly resembling the unit inputs statistic. The most common assumption on the distribution of such patterns is that of *uniform white noise* (UWN). The power of a system composed by several functional units is then computed as the sum of their average power estimators.

Landman et al. [2] realized that for signal processing systems operating with 2's complement numbers, the input probability distribution is *not* UWN and the simple UWN assumption can produce large errors. They concluded that a *dual bit type* (DBT) model is a more accurate representation of signal statistics. The least significant bits have an activity pattern very close to white noise, while the most significant bits (sign bits) have high correlation and cannot be modeled as UWN. The model proposed in [2] takes input statistics into account by increasing the number of fitting coefficients to be obtained during unit characterization. The main limitations of the DBT approach are i) the need of human knowledge for formulating basic models ii) the degradation of the accuracy to simple average power estimate when the sign bits are a small fraction of the inputs. We address both the limitations and we propose a black-box general model for power estimation.

3 Linear regression models

Consider a functional unit with n inputs and m outputs. Assume that the circuit is stable at t_1 and t_2 ($t_2 > t_1$), and that an input transition occurs in the time interval $T = [t_1, t_2]$. We denote by p the power consumption of the circuit in the time period T. Our goal is to find a *black-box pattern dependent model* of p using only boundary information (*i.e.*, the knowledge of the inputs and outputs of the unit at time t_1 and t_2).

To this purpose, we take T equal to the time period between subsequent input patterns, and we follows two simple observations: *i*) in a CMOS combinational circuit, some input has to switch in order to dissipate power, *ii*) the presence of switching outputs corresponds to some internal activity. Moreover, patterns with high input-outputs activity usually lead to higher power dissipation than patterns with lower activity. Obviously this is not always true also because the transitions of different signals may have a different impact on the dissipated power.

We approximate the power dissipation in the circuit by means of a linear regression model based on its input-output activity. The input (output) activity is represented by a vector of Boolean variables $\mathbf{i} = (i_1, i_2, ..., i_n)$ ($\mathbf{o} = (o_1, o_2, ..., o_m)$) taking value 1 when there is a transition on the corresponding input (output) signal. In symbols, our power estimate is

$$p = c_0 + c_1 i_1 + c_2 i_2 + ... + c_n i_n + c_{n+1} o_1 + c_{n+2} o_2 + ... + c_{n+m} o_m \tag{1}$$

where $\mathbf{c} = (c_0, c_1, ..., c_{n+m})$ are fitting coefficients to be determined during characterization.

Obviously, Equation (1) is only a rough approximation: power dissipation is also affected by several other parameters (initial input values, input slopes and signal skews) and its dependence on the I/O activity is *not* exactly linear. On the other hand, signal transitions are the main sources of power consumption and the linear power model is attractive because it is simple and it does not require any knowledge of the actual structure of the unit being modeled.

To determine the coefficients of Equation (1) we need a *sample* of input-output activities and corresponding power consumption. The sample of data collected during the characterization phase can be represented by a pair $(\mathbf{X}, \mathbf{p})$. If s is the sample size, $\mathbf{X}$ is an $s \times (n + m + 1)$ Boolean matrix containing the values taken by the independent variables during characterization (its k-th row being $\mathbf{x}^k = (1, i_1^k, i_2^k, ..., i_n^k, o_1^k, o_2^k, ..., o_m^k)$), while $\mathbf{p}$ is a vector of size s containing the corresponding values of the dependent variable (the k-th element being p^k) obtained from accurate gate-level power simulation.

Given a sample $(\mathbf{X}, \mathbf{p})$, coefficients $\mathbf{c}$ are the unknown of the following system of linear equations:

$$\mathbf{p} = \mathbf{X}\mathbf{c}. \tag{2}$$

Due to the statistic nature of the characterization process, the sample size must be significantly larger than the number of parameters to be characterized. Hence, matrix $\mathbf{X}$ has many more rows than columns and the linear system is overdefined. The vector $\mathbf{c}$

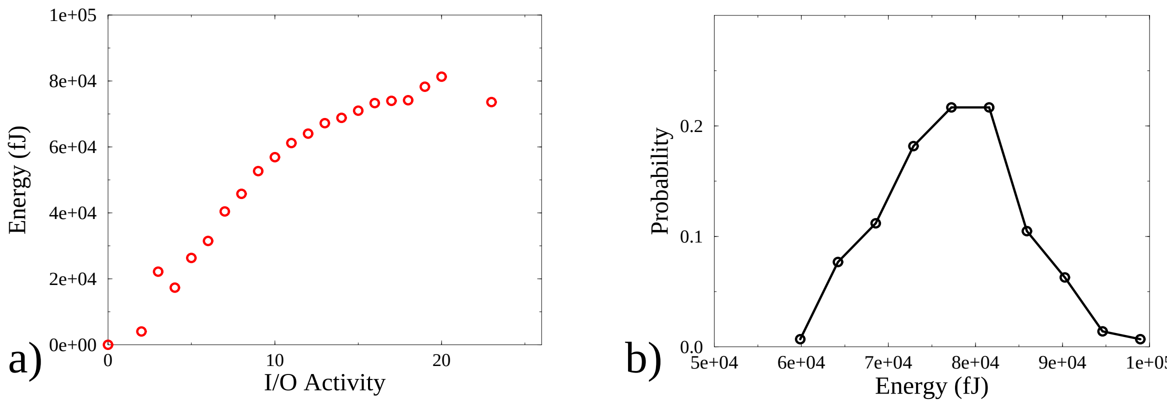

Figure 1: a) Correlation between the I/O activity of an 8-bit carry-lookahead adder and its energy-per-cycle consumption. b) Bell-shaped distribution of the energy consumption of the same circuit due to input transitions corresponding to the same activity vectors (namely, $\mathbf{i} = (0011001100011011), \mathbf{o} = (00101001)$).

giving the minimum mean square error among all possible linear estimators of $\mathbf{p}$ can be obtained from (2) using well-known techniques of least squares fitting [4]. An important property of the least squares linear model is that it always produces an estimate of p with the same average value as the average value of p in the sample used for fitting. Therefore it is guaranteed to perform at least as well as an average value approximation.

3.1 Model validity

In this subsection we check the validity of the linear regression model by discussing the simplifying assumptions we made to construct it.

The first assumption to be checked is that there is correlation between the input-output switching activity and power dissipation. We performed several tests: a typical result is shown in Figure 1.a where the power dissipation is plotted as a function of the total input output activity (*i.e.*, the number of inputs and outputs switching) for an eight-bit carry-lookahead adder. It is apparent that in this case there is good correlation between power consumption and input-output activity. This result is not general, but we experimentally found that it holds for a large set of circuits with functionality ranging from random logic to arithmetic operators. Moreover, the proposed regression model provides a deeper insight than the model used in Figure 1.a in that it accounts for the activity of single inputs and outputs.

The second issue is the robustness of the linear model in presence of the "noise" made by the variation of parameters that do not take part in the model (such as the initial state of the input signals). An important property of the least squares equation is that it provides the optimum fit in a statistical sense. If the dependent variable can be seen as the superposition of a deterministic variable (function of the independent variables) and a random noise with Gaussian distribution, it can be shown that the least squares fit maximizes the probability that for a given value of the independent variables the dependent noisy variable has the value predicted by the least squares solution. We checked the Gaussian hypothesis by plotting the distribution of power dissipation obtained for several input transitions corresponding to the same configurations of $\mathbf{i}$ and $\mathbf{o}$. An example probability distribution for the same adder mentioned before if shown in Fig. 1.b: the bell-shaped curve closely resembles a Gaussian distribution. Again, we do not claim the generality of this result, but our tests show that it holds for a large class of circuits.

Finally, the last hypothesis to be tested is the linearity of the model. Unfortunately, power is not a linear function of the input-output switching activity. The linear model has been chosen because the theory of linear regression is well-established, and it does not require any knowledge of the internal structure of the circuit.

However, trying to fit a non-linear relationship with a linear model may cause sizable errors. Moreover, a gross aspect of non linearities is that the unit may have different modes of operation, with completely dissimilar power consumption.

4 Advanced regression models

The inputs of large logic units can often be grouped into two classes: *control inputs* and *data inputs*. Control inputs have very strong influence on the behavior of the units, because they select different modes of operation. On the other hand, while high activity on data inputs usually correlates well with high power dissipation, such behavior is not observed for control inputs. From this observation, it comes that control inputs can be used to *select among different regression equations*. Given a control variable, and a sample, we split the sample in two subsets, one for each value of the control variable. On the two sub-samples we then compute two new linear regression models.

The main advantage of this procedure is intuitively clear. If the behavior of the logic unit changes radically for different values of the control variable, a single regression model will attempt to find a linear fit between two widely spaced clusters of data. As a result, the fitting will not be satisfactory for neither of the two clusters. If we split the data, and we separately fit the two clusters, much better results are obtained. The

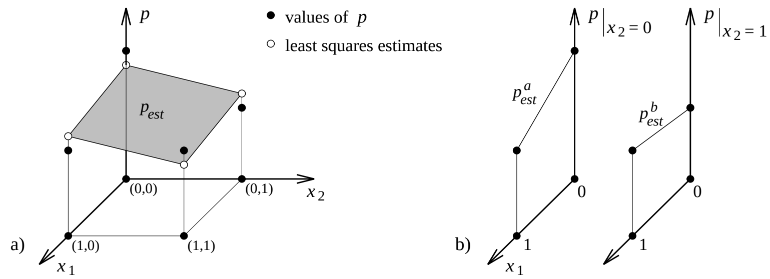

Figure 2: a) Least squares linear approximation of a non-linear function of two Boolean variables. b) Exact fitting of the same function using two linear equations of variable x_1. The value of x_2 is used to switch between the two models.

effectiveness of model splitting is illustrated in Fig. 2 for a two variable function.

This reasoning can be extended to multiple control variables in a recursive fashion. Once we have split the data in two clusters, we can further split if other control variables can be found in the partial models. The structure generated by the recursive splitting is called *regression tree*. The internal nodes of the tree are labeled with the control variables on which we split the model, while the leaves correspond to regression equations with $n + m - d$ independent variables, where d is the depth of the tree. The number of leaves is exponential in the depth of the tree. Consequently, the splitting procedure must be limited to a small number of input variables.

Notice that, in principle, model splitting also addresses non-linearities. A function p of Boolean variables x_1, x_2, ...,x_n is non-linear if and only if some of the independent variables (say x_i) affects not only the value of p, but also the dependence of p on some other variable (say x_j). In other words, x_i plays the role of a control variable. Accuracy can then be improved by using two different regression models for the two values of x_i.

4.1 Splitting criterion

Since our goal is a black-box modeling procedure, we need an automatic splitting criterion based on boundary information. To this purpose we use a statistical approach that can be outlined as follows. i) The global regression model is computed. ii) For each independent variable x_i, the proportion of variance σ_i^2 of the dependent variable y due to x_i is computed. iii) The independent variable with the largest σ_i^2 is chosen for splitting (if σ_i^2 is above an user-defined *splitting threshold*).

The rationale behind this procedure is quite simple. The variance σ_i^2 is high if a change in the value of x_i is associated to a wide variation of y (in average). In other words, if the independent variable x_i selects between two radically different behaviors of the unit, the variance of y due to x_i will be significant.

The advantage of using a statistical method to select the splitting variables is two-fold. No human knowledge is required to steer the characterization process, and the method may be also applied to units with no evident control signals, in order to isolate behaviors with good linearity characteristics.

5 Experimental results

Circuit			Average		Lin.Reg.		Reg.Tree 1		Reg.Tree 2	
name	Ins.	Outs	RMSE	AVGE	RMSE	AVGE	RMSE	AVGE	RMSE	AVGE
alu2	10	6	0.441		0.346		0.335		0.291	
			1.154	0.903	0.484	0.138	0.501	0.197	0.510	0.180
alu4	14	8	0.388		0.294		0.275		0.260	
			1.042	0.762	0.518	0.072	0.549	0.147	0.521	0.119
c17	5	2	0.701		0.422		0.378		0.376	
			1.786	1.325	0.695	0.111	0.686	0.015	0.660	0.070
c432	36	7	0.365		0.207		0.199		0.191	
			1.128	0.849	0.390	0.086	0.385	0.071	0.403	0.122
count	35	16	0.337		0.232		0.227		0.221	
			1.362	1.181	0.428	0.136	0.421	0.103	0.401	0.073
decod	5	16	0.607		0.374		0.315		0.301	
			1.683	1.231	0.636	0.107	0.549	0.100	0.458	0.031
parity	16	1	0.204		0.174		0.164		0.163	
			0.693	0.570	0.382	0.224	0.397	0.251	0.405	0.266
pcle	19	9	0.442		0.364		0.344		0.323	
			1.307	1.038	0.602	0.136	0.605	0.178	0.578	0.113
fastdiv	17	9	0.462		0.364		0.333		0.331	
			1.193	0.729	0.677	0.050	0.666	0.076	0.685	0.086
mult	17	16	0.287		0.263		0.257		0.251	
			0.781	0.596	0.463	0.164	0.459	0.162	0.445	0.132
sqrt	9	4	0.366		0.272		0.269		0.255	
			1.110	0.807	0.496	0.053	0.507	0.112	0.510	0.121

Table 1: Results and comparison for different behavioral power models

We tested our methodology on a set of benchmarks from the MCNC suite and on arithmetic units generated with Synopsys' DesignWare. Notice that even if we have some partial knowledge about the benchmark interface and size, we do not know their internal structure (often we do not even know the functionality). This is the ideal testing environment for our procedure: we want to automatically generate power models for library units without using any knowledge on their structure.

The data on power dissipation has been obtained with PPP [5], an accurate gate-level power simulator that has been reported to produce estimates within 5% from electric simulation (for library-based design in CMOS technology). Notice that electric simulation of our benchmarks would have required an excessively large amount of computation time, thus the availability of a fast and accurate power simulation tools is paramount for model building.

For each circuit in the table we generated a large sample of input patterns and power dissipation. The input patterns used for model building are uniformly distributed and independent. In a different design environment, typical usage trace could be used. We build the regression model using linear regression and tree regression with depth one and two (*i.e.*, two and four leaves). The regression models are compared to the simple estimator given by the average power on the sample (*i.e.*, a pattern-insensitive estimate equal to the average value).

Two different error measures are reported: the relative root mean square error $RMSE$ ($RMSE = \sqrt{MSE}/AVG$, where AVG is the average power on the test sample) and $AVGE$, the relative error on the average ($AVGE = |AVG_{model} - AVG|/AVG$). While $RMSE$ provides information on how well the pattern dependence of power dissipation is modeled, $AVGE$ is a measure of the accuracy in the estimation of the average power.

The results are shown in Table 1. First, we estimated the accuracy of the models on a test sample composed by input vectors randomly chosen from the large sample used in characterization. In this case only the $RMSE$ is significant, because all models give the same (correct) average power estimate by construction and $AVGE$ is not significantly different from zero. It can be seen that the regression tree approach leads to models with improved quality compared to linear regression and constant model.

In the second experiment we generated a new set of input vectors with completely different statistical characteristics from the vectors used for characterization: the switching activity was much reduced (from .5 to .2) and some correlation was randomly introduced between inputs. In this case both $RMSE$ and $AVGE$ are significant. The performance of the constant model is unacceptably degraded, both in average and instantaneous power estimate. In contrast, the performance and robustness of the linear regression model for average power estimation is generally good. Unfortunately, the $RMSE$ is quite high, proving that linear models do not perform well as instantaneous power estimators.

Although it is clear that linear regression outperforms the simple average power pattern independent model, the choice between regression tree and standard linear regression is not straight-forward. It appears that the regression tree is superior to linear regression when the usage patterns are similar to the characterization patterns. If this is not true, standard linear regression leads the best results.

6 Conclusions and Future work

In this work we discussed the theory and the results of linear regression models for power dissipation of combinational hard macros. Our method does not rely on any assumption on data representation and signal probability distribution. No human knowledge is needed for providing an initial model. Our methodology is particularly well suited for design methodologies based on automatic synthesis and standard macro libraries. The linear model has limited accuracy for instantaneous power, but it is remarkably robust and sufficiently accurate for average power estimation. Regression models represent a noticeable improvement with respect to single-parameter power models and are widely applicable.

We incorporated linear regression models in PPP, a logic simulation engine for power estimation based on Verilog XL. PPP provides guidance to the designer during the phases of the design process, from behavioral simulation to gate-level optimization and validation.

References

[1] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE J. of Solid State Circuit*, vol. 29, no. 6, pp. 663–670, 1994.

[2] P. Landman and J. Rabaey, "Architectural power analysis, the Dual Bit Type method," *IEEE Transaction on VLSI Systems*, vol. 3, no. 2, pp. 173–187, 1995.

[3] R. S. Martin and J. Knight, "Power-Profiler: optimizing ASICs power consumption at the behavioral level," in *Proc. of Design Automation Conf.*, pp. 42–47, 1995.

[4] B. L. Bowerman and R. T. O'Connell, *Linear statistical models - An applied approach.* PWS-Kent, 1990.

[5] A. Bogliolo, L. Benini, and B. Riccò, "Power Estimation of Cell-Based CMOS Circuits," *to appear in Proceedings of the Design Automation Conference*, 1996.