

Parameterized RTL Power Models for Soft Macros

Alessandro Bogliolo, Roberto Corgnati, Enrico Macii, *Member, IEEE*, and Massimo Poncino

Abstract—We propose a new power macromodel for usage in the context of register-transfer level (RTL) power estimation. The model is suitable for reconfigurable, synthesizable, soft macros because it is parameterized with respect to the input data size (i.e., bit width) and can also be automatically scaled with respect to different technology libraries and/or synthesis options. The power model is precharacterized once and for all for each soft macro and then adapted to each specific instance by means of a single additional experiment to be performed by the end user. No intellectual-property disclosure is required for model scaling. The proposed model is derived from empirical analysis of the sensitivity of power consumption on input statistics, input data size, and technology. The experiments prove that with limited approximation, it is possible to decouple the effects on power of these three factors. The proposed solution is innovative since no previous macromodel supports automatic technology scaling and yields average estimation errors around 10%.

Index Terms—Low-power design, low-power dissipation, power-consumption model.

I. INTRODUCTION

HIGH-LEVEL power estimation has become an essential task in modern design flows. Several researchers have addressed this issue, mainly at the register-transfer level (RTL), proposing solutions that are based on the construction of abstract power *macromodels*, starting from existing or newly synthesized implementations [1], [2]. Among others, *regression-based* [3], [4] and *table-based* [5], [6] macromodels are available.

Today's designs are increasingly based on the use of *soft macros*. Such macros are library macrocells for which only a parametric, synthesizable, and technology-independent description is provided. They are much more flexible than *hard macros* because they can be mapped onto different technology libraries and their structural characteristics (e.g., bit width) can be adapted to those of the data path. Flexibility is paid in terms of uncertainty when estimating the impact of a soft macro on the design cost metrics. For power, the lack of a presynthesized implementation does not allow accurate precharacterization based on low-level simulation.

The power consumption of a soft macro can be viewed as a function of three parameters: 1) its run-time input statistics; 2) its structural and functional parameters; and 3) its implementa-

tion, i.e., the library it is mapped onto and the synthesis script used for mapping.

Most of the macromodels proposed in the literature are focused on hard macros; that is, they are typically conceived for modules with a fixed architecture and implementation. Therefore, they capture only the dependence of power on the input statistics and do not provide any flexibility to adapt the estimates to different structural parameters and technologies.

Only a few parameterized power models have been proposed that depend on the bit width of the input data [1], [7]. In these techniques, the relation between bit width and power is modeled by specific analytical functions that need to be defined for each class of macros. On the other hand, no support is provided to adapt the power models to different synthesis scripts and technology libraries.

In this paper, we follow a different approach. We first conduct an experimental analysis of the dependence of power on the three sets of parameters discussed above. The main result of the analysis is that such parameters have an almost disjoint effect on power; therefore, they can be independently modeled with low accuracy loss. We thus propose a power modeling technique for soft macros based on the concept of *scaling*: A power model is first built and precharacterized for a reference instance of the macro (capturing only the dependence on input statistics) and then scaled to adapt to the actual input bit width and technology. Intellectual property (IP) issues involved in model characterization and scaling are also discussed and addressed.

The proposed approach was validated on combinational and sequential soft macros taken from the Synopsys DesignWare Library, showing that scaling can be effectively applied to adapt to soft macros traditional power models originally conceived for hard macros. The accuracy provided by the scaled power models is comparable with that provided by *ad hoc* models of the same type, directly characterized for the target parameters' setting and technology.

II. PRELIMINARY ANALYSIS

Soft macros may have both size and functional parameters. Size parameters are used to adapt the bit width of the input-output data of the macro to that of the data path, while functional parameters (when present) are used to select particular functional or encoding options. For our purposes, we treat size and functional parameters in two different ways. Macros with different functional parameters are regarded as different macros for which different power models can be independently built and characterized, while macros differing only for the size parameters are regarded as different instances of the same soft macro for which a unique scalable model has to be provided. There are three main practical motivations for this choice: First,

Manuscript received May 12, 2000; revised August 18, 2000. This work was supported by the Commission of European Communities DG-III under ESPRIT Project 26796 "PEOPLE."

A. Bogliolo is with the University of Ferrara, Ferrara 44100, Italy (e-mail: abogliolo@ing.unife.it).

R. Corgnati, E. Macii, and M. Poncino are with Politecnico di Torino, Torino 10129, Italy (e-mail: corgnati@athena.polito.it; enrico@athena.polito.it; poncino@athena.polito.it).

Publisher Item Identifier S 1063-8210(01)06664-1.

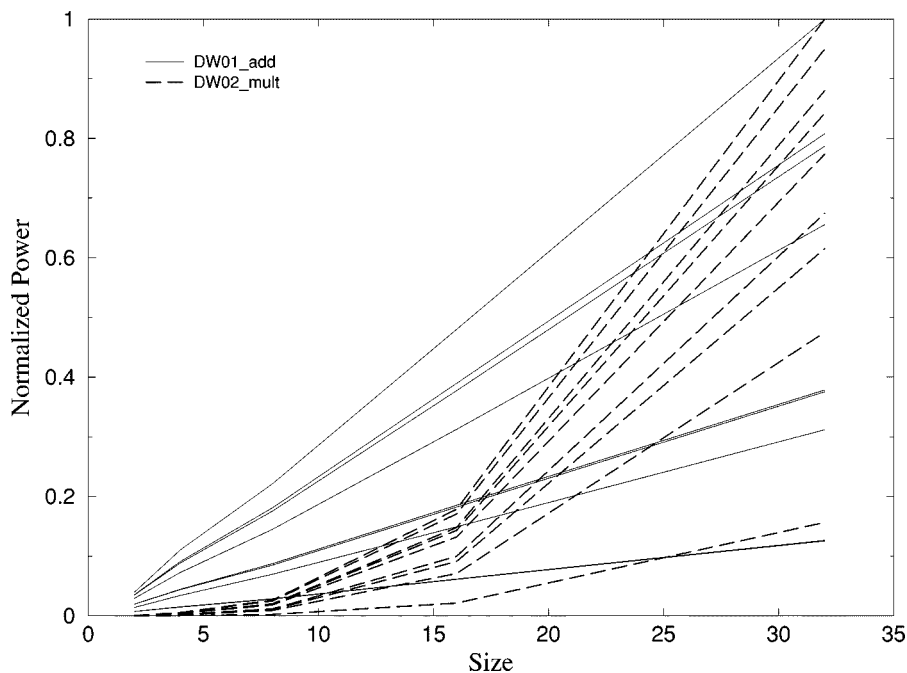


Fig. 1. Power as a function of data bit width.

size parameters are common to all macros, while functional parameters are particular to each macro so that their effect cannot be generalized; second, size parameters may take a large number of different values (making it highly inefficient to characterize independent power models for each assignment) while functional parameters (if any) usually select among a few options; third, functional parameters may completely change the functionality of the macro, while size parameters do not. In practice, we propose a power modeling approach for soft macros with only size parameters. If functional parameters are also present, the proposed approach has to be independently applied for each assignment of the functional parameters.

We represent the power consumption of a soft macro as a function P of its boundary statistics ($stat$), its input–output bit width ($size$) and its implementation ($tech$). Each point in the input space of $P(stat, size, tech)$ is associated with a different configuration of the independent variables, originating a different power consumption.

For a given macro, the actual value of P in a given point $(stat, size, tech)$ can be obtained by means of synthesis and simulation. An instance of the macro with $size = \overline{size}$ has to be synthesized, mapped onto technology library $tech = \overline{tech}$ and simulated for power at the gate-level with input statistics $stat = \overline{stat}$. We studied the dependence of power on $stat$, $size$ and $tech$ for several soft macros (taken from the Synopsys DesignWare Library) by sampling the input space and observing the value of P in each sampling point. Synopsys DesignCompiler, VSS, and DesignPower were used for synthesis, simulation, and power analysis, respectively.

The dependence on size is plotted in Fig. 1 for an adder and a multiplier with a unique size parameter representing the bit width of the two input ports. Power values for each macro are normalized to the highest obtained. Piecewise linear curves are drawn by connecting points associated with the same input statistics so that different curves refer to different input statistics.

All results in the plot are obtained by using the same technology library (namely, Synopsys `class.db`) and synthesis options, i.e., they correspond to sampling points laying on a plane $tech = const$ in the input space.

From the plot we observe the following.

- 1) The relation between bit width and power depends on the macro (for instance, it is almost linear for the adder, while it is nearly quadratic for the multiplier, as expected and already observed [1]).
- 2) Curves of the same family (i.e., adder, multiplier) never intersect and their relative ratio is almost independent of the bit width.

Observation 1 suggests that there is no general formula relating power consumption to bit width. Observation 2 suggests that the dependence on the bit width can be decoupled from the dependence on the input statistics.

To verify the correctness of the assumption on disjoint dependence of power from input statistics and bit width, we analyze the behavior of the ratio (R_1) between the value of P in points differing only for the $size$ parameter. In symbols

$$R_1(stat, size_1, size_2, tech) = \frac{P(stat, size_1, tech)}{P(stat, size_2, tech)}.$$

R_1 represents the ratio between the power consumption associated with points having the same $(stat, tech)$ coordinates on two parallel planes of equations $size = size_1$ and $size = size_2$.

Fig. 2 shows the value of R_1 for different assignments of $size_1$, $size_2$, $stat$ and $tech$. Each curve is associated with a $(size_1, size_2)$ pair, while labels a to i on the x axis denote different configurations of $stat$ and $tech$.

We observe that for fixed values of $size_1$ and $size_2$, R_1 is almost independent of $stat$ and $tech$, so that it can be regarded as a property of the pair of parallel planes ($size = size_1$ and

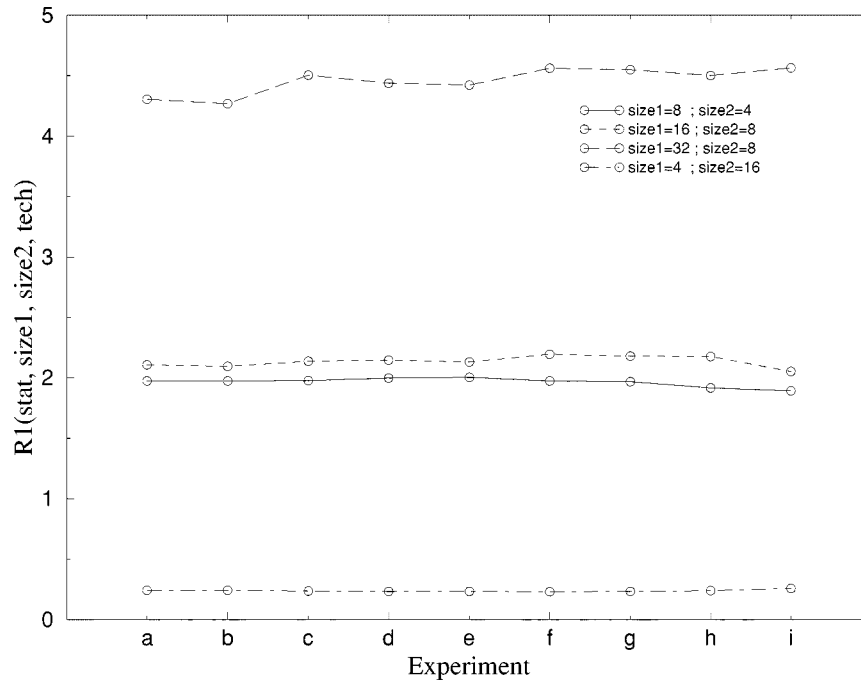


Fig. 2. Plot of R_1 as a function of $stat$ and $tech$ for an adder.

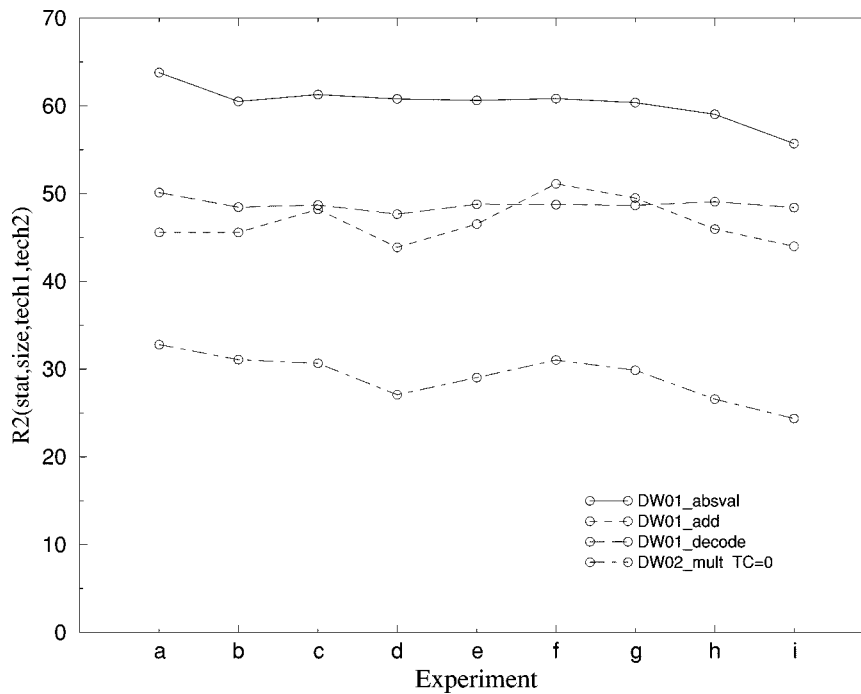


Fig. 3. Plot of R_2 as a function of $stat$ and $size$ for different macros.

$size = size_2$) that does not depend on the position of the points on the plane. The assumption of disjoint dependence is thus validated, suggesting a partitioning of the modeling task into two subtasks, separately focusing on the dependence of power on $size$ and on $stat$ and $tech$.

A second ratio, R_2 , can be defined and studied to analyze the dependence of power consumption on technology and synthesis

$$R_2(stat, size, tech_1, tech_2) = \frac{P(stat, size, tech_1)}{P(stat, size, tech_2)}$$

It expresses the impact of technology changes from $tech_2$ to $tech_1$. We used the Synopsys `class.db` as reference library ($tech_2$), while we used either a simple three-gate library (denoted by `simple`) or a commercial library containing over 200 cells (denoted by `industrial`) as custom library ($tech_1$).

Fig. 3 plots several values of $R_2(stat, size, tech_1, tech_2)$ as a function of $stat$ and $size$, computed for $tech_1 = industrial$ and $tech_2 = class.db$. Interestingly, for each macro the ratio is weakly affected by input statistics and size parameters, sug-

gesting that the dependence on technology can be modeled independently of that on *size* and *stat*.

To better quantify the error possibly due to the assumption of disjoint dependence of power on *stat*, *size* and *tech*, we have evaluated the *relative* standard deviation of R_1 and R_2 over all our experiments. Results, averaged over all the considered macros, are 7.5% for R_1 and 15.4% for R_2 . Such low values justify the modeling strategy discussed in Section III.

III. SCALABLE POWER MODELS

According to the analysis of Section II, we model power consumption of a soft macro as a three-term product

$$P(\textit{size}, \textit{stat}, \textit{tech}) = P_{\text{ref}}(\textit{stat})S_1(\textit{size})S_2(\textit{tech}) \quad (1)$$

where $P_{\text{ref}}(\textit{stat})$ is a power model for a reference implementation of the soft macro, synthesized with size $\textit{size}_{\text{ref}}$ and technology library $\textit{tech}_{\text{ref}}$. In other words

$$P(\textit{size}_{\text{ref}}, \textit{stat}, \textit{tech}_{\text{ref}}) = P_{\text{ref}}(\textit{stat}).$$

$S_1(\textit{size})$ and $S_2(\textit{tech})$ are scaling factors that contribute to adapt $P_{\text{ref}}(\textit{stat})$ to all possible realizations obtained with different bit widths, tech libraries, and synthesis directives.

A. Reference Power Model

When building $P_{\text{ref}}(\textit{stat})$, the reference instance of the soft macro can be regarded as a hard macro for which a power model can be pre-characterized by means of accurate low-level simulations. Hence, any activity-sensitive RTL power model traditionally used for hard macros [8] can be used to model $P_{\text{ref}}(\textit{stat})$. For our experiments, we chose the three-dimensional (3-D) lookup table power model (hereafter called Tab3D) proposed by Gupta *et al.* [6] because of its simplicity and generality (its applicability to sequential macros has been recently tested [9]). As boundary statistics for the Tab3D model we use three probabilities: P_{in} , the average input signal probability, D_{in} , the average input transition probability, and D_{out} , the average output transition probability. The dependence of P on P_{in} , D_{in} and D_{out} is represented by means of a 3-D lookup table whose entries represent the power values associated with a triplet of boundary statistics

$$P_{\text{ref}}(\textit{stat}) = \text{Tab3D}(P_{\text{in}}, D_{\text{in}}, D_{\text{out}}).$$

The Tab3D model allows the designer to span the tradeoff between accuracy and model dimension simply by changing the discretization step in the three dimensions. The discretization step used for our experiments was 0.1 for P_{in} and D_{in} ; no discretization was used on the D_{out} axis (the values of D_{out} obtained during characterization were stored in the model together with the corresponding power values).

We remark that so far we have used *stat* to denote any n -tuple of boundary statistics needed to evaluate an arbitrary baseline power model $P(\textit{stat})$. Having chosen Tab3D for our experiments, hereafter we will use *stat* to denote the triplet $(P_{\text{in}}, D_{\text{in}}, D_{\text{out}})$ the model is based on.

B. Scaling Factors

Experiments have shown that the relation between bit width and power of a macro depends on its functionality. Hence, it is

not possible to find a general analytical formula for $S_1(\textit{size})$. On the other hand, according to our modeling assumptions, ratio R_1 is almost independent of *stat* and *tech*. Hence, we can characterize $S_1(\textit{size})$ for a fixed, arbitrary configuration $(\textit{stat}_{\text{ref}}, \textit{tech}_{\text{ref}})$ of $(\textit{stat}, \textit{tech})$ and use it all over the input space. A model for $S_1(\textit{size})$ is provided by its definition

$$S_1(\textit{size}) = \frac{P(\textit{stat}_{\text{ref}}, \textit{size}, \textit{tech}_{\text{ref}})}{P(\textit{stat}_{\text{ref}}, \textit{size}_{\text{ref}}, \textit{tech}_{\text{ref}})} \quad (2)$$

where $\textit{size}_{\text{ref}}$ and $\textit{tech}_{\text{ref}}$ are the size and technology of the reference implementation for which $P_{\text{ref}}(\textit{stat})$ was characterized, while $\textit{stat}_{\text{ref}}$ is the fixed configuration of boundary statistics. For our experiments we used $P_{\text{in}} = D_{\text{in}} = 0.5$ and the resulting value of D_{out} obtained from simulation. Since *size* usually takes a limited number of integer values (e.g., 4, 8, 16, 32, 64), $S_1(\textit{size})$ can be precomputed for typical sizes and stored in a table. Characterization entails, for each value of size, the evaluation of $P(\textit{stat}_{\text{ref}}, \textit{size}, \textit{tech}_{\text{ref}})$, that, in turn, requires synthesis and simulation.

Similar considerations apply to technology scaling factor S_2 . By definition, it can be computed as

$$S_2(\textit{tech}) = \frac{P(\textit{stat}_{\text{ref}}, \textit{size}_{\text{ref}}, \textit{tech})}{P(\textit{stat}_{\text{ref}}, \textit{size}_{\text{ref}}, \textit{tech}_{\text{ref}})} \quad (3)$$

where $\textit{size}_{\text{ref}}$ and $\textit{stat}_{\text{ref}}$ are the arbitrary assignments of the fixed parameters. However, unlike $S_1(\textit{size})$, $S_2(\textit{tech})$ cannot be statically precharacterized nor tabulated since it is even impossible to enumerate all technology libraries and synthesis scripts a designer may use. Hence, technology scaling has to be dynamically performed by the designer once the target technology and synthesis parameters have been defined.

C. Model Characterization

The characterization procedure we outline in the sequel needs to be carried out once for each soft macro, either by the developer of the library of RTL macros (if the power models are to be distributed together with the library and/or there are intellectual property (IP) issues involved) or by the designer (if the library has not been precharacterized by the developer and it is completely accessible to the designer). The characterization process consists of three steps.

Characterization of P_{ref} :

A reference technology ($\textit{tech}_{\text{ref}}$) is chosen and used to synthesize the macro for a reference bit width ($\textit{size}_{\text{ref}}$). In our experiments, we used Synopsys `class.db` library and $\textit{size}_{\text{ref}} = 8$. Tab3D is then characterized for this reference implementation using input streams with different statistics. We used a sampling step of 0.1 in the P_{in} and D_{in} dimensions and, for every $(P_{\text{in}}, D_{\text{in}})$ pair, we generated 20 input streams of 50 patterns each with different bit-wise statistics. All input streams were simulated using Synopsys VSS and DesignPower to obtain the corresponding values for D_{out} and power.

Characterization of $S_1(\textit{size})$:

The size scaling factor is characterized by synthesizing the soft macro for different configurations of the bit width parameters (ranging from 4 to 32 in our experiments) and simulating each implementation with

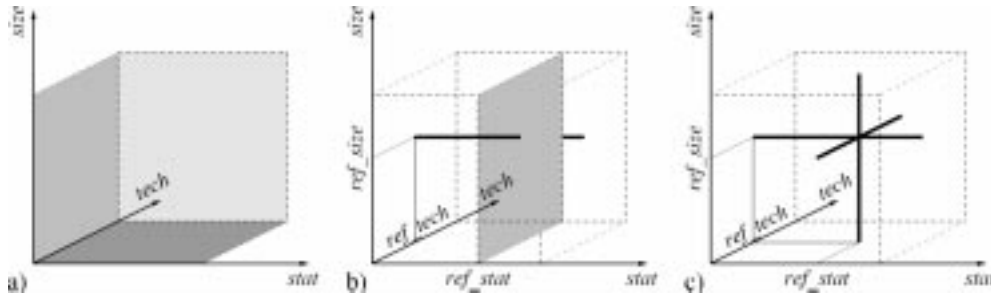


Fig. 4. Input space of the power model of a soft macro. (a) The entire space. (b) Independent exploration of the $stat$ dimension. (c) Independent exploration of the three dimensions.

the reference input stream $stat_{ref}$ (we used $P_{in} = D_{in} = 0.5$ for all input data bits).

Characterization of $S_2(tech)$:

The technology scaling factor cannot be characterized in advance. Rather, it has to be computed by the designer for each target technology. What can be statically precomputed is the denominator $P(stat_{ref}, size_{ref}, tech_{ref})$ that requires a single simulation run of the reference implementation for a given input stream.

D. Model Evaluation

The power estimate for an instance of a soft macro is given by

$$\text{Tab3D}(P_{in}, D_{in}, D_{out})S_1(size)S_2(tech)$$

where

$(P_{in}, D_{in}, D_{out})$ boundary statistics;
 $size$ actual configuration of the size parameters;
 $tech$ target technology.

Model evaluation consists of the following.

Technology Scaling:

The user of the model must first perform technology scaling to adapt the power model to his/her own synthesis flow and technology library ($tech$). To this purpose, an instance of the macro with size $size_{ref}$ has to be synthesized and simulated using the same input stream used during the third step of the characterization process. The resulting power value $P(stat_{ref}, size_{ref}, tech)$ is then used to compute $S_2(tech)$ according to its definition. Notice that $S_2(tech)$ has to be computed once for each soft macro in the RTL library.

Size Scaling:

Scaling factor $S_1(size)$ is precharacterized and stored in a lookup table. In this way, during model evaluation, size scaling only requires a single lookup for each instance of the macro.

Power Evaluation:

Finally, boundary statistics (P_{in} , D_{in} and D_{out}) are collected for each instance of a soft macro during RTL simulation of the entire design and used to evaluate the corresponding Tab3D models.

E. Accuracy versus Efficiency Tradeoff

In principle, a power model for a soft macro could be viewed as a multi-dimensional lookup table addressed by a discretized configuration of $(stat, size, tech)$. Characterization of entry $(\overline{stat}, \overline{size}, \overline{tech})$ would require synthesis of an instance of the macro with $size = \overline{size}$, mapping on technology \overline{tech} and simulation for input statistics \overline{stat} . If we denote by N_{stat} , N_{size} and N_{tech} the number of points on each axis, it can be easily found that the table would have $N_{stat} \cdot N_{size} \cdot N_{tech}$ entries whose characterization would require $N_{stat} \cdot N_{size} \cdot N_{tech}$ simulations and $N_{size} \cdot N_{tech}$ synthesis runs.

Thanks to the quasi-disjoint dependence of power on the three sets of parameters observed in Section II, we modeled power consumption as the product of three terms separately representing the dependence on a single variable. Each term could be modeled as a linear table of N_{stat} , N_{size} and N_{tech} entries, respectively. Characterization of the three-term model needs $N_{stat} + N_{size} + N_{tech}$ simulations and $N_{size} + N_{tech}$ synthesis runs.

The sizeable advantage, in terms of model efficiency, provided by this approach is pictorially shown in Fig. 4: The cube represents the entire input space where the original lookup table of $N_{stat} \cdot N_{size} \cdot N_{tech}$ entries is defined, while the three orthogonal segments of Fig. 4(c) represent the linear sets of $N_{stat} + N_{size} + N_{tech}$ points where the three terms of our power model must be characterized. The value of power in any point $(stat, size, tech)$ of the input space can then be obtained as the product of the orthogonal models characterized on the three segments.

The efficiency improvement is paid in terms of accuracy loss due to the simplifying assumption of complete disjointness of the three effects. To improve accuracy we can reduce the strength of the assumption by using a single scaling factor S modeling the dependence on both size and technology. The resulting model will be

$$P(stat, size, tech) = P(stat)S(size, tech). \quad (4)$$

The efficiency of this modified model is represented in Fig. 4(b); scaling factor S is now defined on a plane covered by a lookup table of $N_{size} \cdot N_{tech}$ entries. Characterization now requires $N_{stat} + N_{size} \cdot N_{tech}$ simulations and $N_{size} \cdot N_{tech}$ synthesis runs.

Characterization of the scaling coefficient $S(size, tech)$ entails the following: 1) synthesis of an instance with target size parameters; 2) mapping of the instance on the target technology;

TABLE I
EXPERIMENTAL RESULTS

Macro	AdHoc		Eq.4		Eq.1		Eq.5(a)		Eq.5(b)	
	Err	Dev	Err	Dev	Err	Dev	Err	Dev	Err	Dev
sub	1.8	1.5	6.2	3.6	10.1	4.8	8.8	4.3	8.8	4.3
absval	3.7	3.5	7.2	5.9	7.5	6.4	8.7	6.6	8.7	6.6
add	2.6	2.3	6.2	4.1	12.0	6.5	19.4	5.9	19.4	5.9
addsub	2.7	2.6	7.7	4.8	10.3	6.8	11.9	7.2	11.9	7.2
decode	2.6	3.1	6.5	5.1	6.3	4.6	22.7	8.6	6.3	5.1
cmp2	4.1	3.8	7.6	6.1	8.2	6.3	8.2	6.1	8.5	5.3
cmp6	3.0	2.8	8.2	5.3	8.9	6.1	11.2	7.7	9.0	6.2
dec	4.6	4.1	7.9	6.2	7.6	6.0	18.1	8.3	8.3	6.2
inc	5.4	5.3	8.9	5.6	11.5	6.2	17.2	6.1	14.2	6.3
incdec	4.7	4.9	7.4	5.5	11.1	6.5	17.3	5.7	15.3	5.4
mult	3.6	4.5	9.4	10.2	37.1	13.4	38.0	14.5	37.1	13.4
u_count	1.7	0.7	2.2	1.4	14.0	3.0	10.1	2.4	10.6	3.2
d_count	0.8	0.7	3.6	3.4	10.5	3.0	17.8	3.9	10.5	3.0
register	0.5	0.3	6.1	3.8	5.9	3.6	18.9	7.2	7.1	4.1
ACC	1.9	1.6	5.3	4.0	11.1	4.9	25.1	4.7	14.3	5.0
Avg.	2.9	2.8	6.7	5.0	11.5	5.6	16.9	6.3	12.7	5.8

and 3) gate-level simulation using the reference input stream. The average power consumption provided by gate-level simulation is $P(stat_{ref}, size, tech)$. The scaling factor is then obtained as

$$S(size, tech) = \frac{P(stat_{ref}, size, tech)}{P(stat_{ref}, size_{ref}, tech_{ref})}.$$

Accuracy improvements are discussed in Section IV.

F. Intellectual Property Protection

If the soft macros are a third-party IP, the power models need to be precharacterized by the developer and made available to the designer together with the functional description of the macros. The underlying assumption behind this “user-provider” paradigm is that model characterization requires complete access to the library, while model evaluation does not require IP critical information.

Unfortunately, this assumption does not hold for technology scaling. $S_2(tech)$ cannot be precharacterized by the library developer since the actual technology parameters can be arbitrarily defined by the designer. Conversely, evaluating $S_2(tech)$ requires synthesis and low-level simulation so that it cannot be performed by the designer if he/she has no access to a synthesizable description of the macro (that is a provider’s IP).

To enable technology scaling while protecting IP we sacrifice some accuracy. The solution we propose is based on the observation that $S_2(tech)$ does not vary too much from a macro to another. Hence, a unique technology scaling model $\bar{S}_2(tech)$ could be used for all macros at the cost of some accuracy loss. On the other hand, a “dummy” macro containing no IP could be freely distributed to the designers to be used for evaluating $\bar{S}_2(tech)$.

From the provider’s point of view, a dummy macro has to be chosen for technology scaling, synthesized with $size_{ref}$, mapped onto $tech_{ref}$ and simulated with a given input stream to obtain the value of $\bar{P}(stat_{ref}, size_{ref}, tech_{ref})$ to be supplied to the end user together with the input stream and with the synthesizable description of the macro. This process has to be performed only once.

From the designer’s point of view, technology scaling entails synthesis and simulation (as described in Section III-D) with two main differences: The dummy macro has to be used instead of the actual macros, and the scaling factor has to be computed only once and then used to scale all models to the actual technology.

The IP conscious power model has the form

$$P(size, stat, tech) = P_{ref}(stat)S_1(size)\bar{S}_2(tech). \quad (5)$$

Its efficiency and accuracy are discussed in Section IV.

IV. EXPERIMENTAL RESULTS

We have benchmarked the accuracy of the proposed power models on combinational and sequential soft macros taken from the Synopsys DesignWare Library, except for macro ACC, a parameterized accumulator consisting of a register and an adder. For each benchmark, we have compared five models.

Ad Hoc Implementation-specific Tab3D models constructed for each instance of the soft macros as if they were *hard macros*.

(4) The model of (4) with a single scaling factor.

(1) The three-term model of (1).

(5a) The IP conscious power model of (5), using a unique dummy macro (namely, the subtractor DW01_sub of size 32) for technology scaling.

(5b) The IP-conscious power model of (5), using four different dummy macros for technology scaling, as explained later in this section.

All models were evaluated for three different technology libraries (i.e., `class.db`, `simple` and `industrial`), four different sizes of the bit width (4, 8, 16, and 32) and 36 different input statistics [9 (P_{in} , D_{in}) pairs were randomly selected, and four input streams, with different bitwise statistics, were generated for each pair].

The resulting power estimates were then compared to those provided by gate-level simulations performed by VSS and DesignPower. Experimental results are reported in Table I in terms

TABLE II
COMPUTATIONAL COST OF THE MODELS

	<i>AdHoc</i>		<i>Eq.4</i>		<i>Eq.1</i>		<i>Eq.5(a)</i>		<i>Eq.5(b)</i>	
	<i>Char</i>	<i>Eval</i>	<i>Char</i>	<i>Eval</i>	<i>Char</i>	<i>Eval</i>	<i>Char</i>	<i>Eval</i>	<i>Char</i>	<i>Eval</i>
# <i>Syn.</i>	48	0	1	47	16	2	16	1/ <i>N</i>	16	1/ <i>n</i>
# <i>Sim.</i>	48000	0	1000	47	1015	2	1015	1/ <i>N</i>	1015	1/ <i>n</i>

of average percentage error (*Err*) and standard deviation (*Dev*) of the percentage error.

The errors reported in the *Ad Hoc* column show the inherent accuracy of the Tab3D model (without scaling). Its average error of 2.9% has been used as a baseline for evaluating the approximation introduced by size and technology scaling. As expected, (4) is the most accurate of the scaled models, with an average percentage error of 6.7%. Equation (1) introduces some additional errors because of the stronger assumption of disjointness of parameters *stat*, *size* and *tech*, yielding an average error of 11.5%. Finally, (5a) trades accuracy for performance and IP protection, providing an average error of 16.9%.

It should be noted, however, that IP protection could also be achieved by using more than one dummy macro for technology scaling, in order to improve model accuracy. In fact, by analyzing the individual entries of Table I, we observe that model (5) provides good accuracy when applied to macros with high affinity to the one used for scaling (e.g., the adder), while errors tend to be larger for macros with a different structure (e.g., the decoder). More specifically, we were able to reduce the average error of model (5) at 12.7% [thus, making it comparable in accuracy with model (1)] by aggregating the benchmarks into four categories according to the functions they implement (*arithmetic_1*, *arithmetic_2*, *logic*, *sequential*) and by using a different dummy macro for each category. Results are collected in column (5b) of the table.

Estimation performance are reported in Table II in terms of number of synthesis and simulation runs required for characterization (*Char*) and evaluation (*Eval*) for each model. Numbers refer to the case of a macro for which a Tab3D model of 1000 entries has to be characterized, 16 configurations of the size parameters can be specified and three different technology libraries can be used for mapping. In symbols: $N_{stat} = 1000$, $N_{size} = 16$, $N_{tech} = 3$. The *Eval* column gives the total number of synthesis and simulation experiments required to evaluate the models for all possible instances of the same macro (entries $1/N$ and $1/n$ denote that a single synthesis run is required to scale either all the N macros in a library or a subset of n of them).

V. CONCLUSION

We have proposed a parameterizable power macromodel suitable for RTL soft macros. The main feature of the model is its capability of self adapting to different bit widths of the data inputs, synthesis options and/or technology libraries. The experiments have shown that model scaling provides an average error of 6.7% to be compared with the intrinsic average error of 2.9% resulting from a power model ideally characterized for each in-

stance of the soft macro. Scaling requires a single experiment to be performed by the end user to adapt the models to his/her synthesis flow. IP protection, as imposed by commercial synthesizable macro libraries, can be guaranteed by performing the scaling experiments on a public-domain "dummy" macro.

REFERENCES

- [1] P. Landman and J. Rabaey, "Architectural power analysis: The dual-bit type model," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 173–187, Feb. 1995.
- [2] A. Raghunathan, S. Dey, and N. Jha, "Register-transfer level estimation techniques for switching activity and power consumption," in *ICCAD'96: Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 1996, pp. 158–165.
- [3] Q. Qiu, Q. Wu, C.-S. Ding, and M. Pedram, "Cycle-accurate macro-models for RT-level power analysis," in *ISLPED'97: ACM/IEEE Int. Symp. Low-Power Electronics and Design*, Monterey, CA, Aug. 1997, pp. 125–130.
- [4] A. Bogliolo, L. Benini, and G. De Micheli, "Adaptive least mean square behavioral power modeling," in *EDTC'97: IEEE European Design and Test Conf.*, Paris, France, Mar. 1997, pp. 404–410.
- [5] H. Mehta, R. M. Owens, and M. J. Irwin, "Energy characterization based on clustering," in *DAC'96: ACM/IEEE Design Automation Conf.*, Las Vegas, NV, June 1996, pp. 702–707.
- [6] S. Gupta and F. Najm, "Power macromodeling for high level power estimation," in *DAC'97: ACM/IEEE Design Automation Conf.*, Anaheim, CA, June 1997, pp. 365–370.
- [7] G. Jochens, L. Kruse, E. Schmidt, and W. Nebel, "A new parameterizable power macro-model for datapath components," in *DATE'99: IEEE Design Automation and Test Eur.*, Munich, Germany, Mar. 1999, pp. 29–36.
- [8] P. Landman, "High-level power estimation," in *ISLPED'96: ACM/IEEE Int. Symp. Low-Power Electronics and Design*, Monterey, CA, Aug. 1996, pp. 29–35.
- [9] A. Bogliolo, E. Macii, V. Mihailovici, and M. Poncino, "Combinational characterization-based power macro-models for sequential macros," in *PATMOS'99: 9th Int. Workshop Power and Timing Modeling, Optimization and Simulation*, Kos Island, Greece, Oct. 1999, pp. 293–302.

Alessandro Bogliolo received the Dr.Eng. degree in electrical engineering and the Ph.D. degree in electrical engineering and computer science from the University of Bologna, Bologna, Italy, in 1992 and 1998, respectively.

From 1992 to 1999, he was with the Department of Electronics and Computer Science (DEIS), University of Bologna. From 1995 to 1996, he was a Visiting Scholar at the Computer Systems Laboratory (CSL), Stanford University, Stanford, CA. Since then he has cooperated with the research group of Prof. De Micheli, Stanford, CA. In 1999, he joined the Department of Engineering, University of Ferrara, Ferrara, Italy, as an Assistant Professor. His research interests are in the area of computer-aided design of digital circuits and systems, with emphasis on low-power design, run-time power management, intellectual-property protection, and signal integrity.

Roberto Corgnati received the Dr.Eng. degree in electrical engineering from Politecnico di Torino, Italy, in 1998. He is currently working toward the Ph.D. degree in computer engineering at the same institution.

His research interests are in the development of algorithms, methodologies, and tools for the design of low-power digital systems.

Enrico Macii (M'92) received the Dr.Eng. degree in electrical engineering from Politecnico di Torino, Italy, the Dr.Sc. degree in computer science from Università di Torino, and the Ph.D. degree in computer engineering from Politecnico di Torino, in 1990, 1991, and 1995, respectively.

From 1991 to 1994, he was an Adjunct Faculty Member with the University of Colorado at Boulder. Currently, he is an Associate Professor of computer engineering with the Politecnico di Torino. He is an Associate Editor of the *ACM Transactions on Design Automation*. His research interests include several aspects of the computer-aided design of integrated circuits and systems, with particular emphasis on synthesis, optimization, and formal verification.

Dr. Macii is an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN. He was the technical program Co-Chair of the IEEE Alessandro Volta Memorial Workshop on Low-Power Design in 1999, the technical program Co-Chair and the General Chair of the ACM/IEEE International Symposium on Low-Power Electronics and Design in 2000 and 2001, respectively.

Massimo Ponicino received the Dr.Eng. degree in electrical engineering and the Ph.D. degree in computer engineering, both from Politecnico di Torino, Italy, in 1989 and 1993, respectively.

From 1993 to 1994, he was a Visiting Faculty Member at the University of Colorado at Boulder. Currently, he is an Assistant Professor with Politecnico di Torino. His research interests include synthesis, verification, simulation, and testing of digital circuits and systems.