

# A Multilevel Engine for Fast Power Simulation of Realistic Input Streams

Luca Benini, Giovanni De Micheli, *Fellow, IEEE*, Enrico Macii, Massimo Poncino, and Riccardo Scarsi

**Abstract**—Power estimation for validation and sign-off is a critical step in the design process. In this phase, accuracy is a key requirement, but there are hard constraints on the time that can be dedicated to power estimation. Moreover, it is important to estimate the power dissipated by the system while running typical applications, i.e., extremely long streams of validation patterns provided by the designer.

The power dissipated by digital systems under realistic input stimuli is not accurately described by a single average value, but by a waveform that shows how power consumption varies over time as the system responds to the inputs. In this paper, we face the problem of obtaining accurate power waveforms for combinational and sequential circuits under typical usage patterns. We propose a multilevel simulation engine that achieves high accuracy in estimating the time-domain power waveform, as well as the average power with high computational efficiency.

**Index Terms**—Design automation, digital circuits, power estimation, simulation.

## I. INTRODUCTION

THE ESTIMATION of power dissipation for digital CMOS circuits is a central issue for designers and system engineers. In the early phases of the design process, a large number of solutions is explored and evaluated. Estimation techniques for design exploration operate on incomplete or partially specified information and must be fast because they are usually applied to several design alternatives [1]. Clearly, these requirements can be satisfied at the price of a reduced accuracy, that can be tolerated as long as estimation provides reliable information on the relative power dissipation of competing options.

In this paper, we focus on another aspect of the power estimation problem. After architectural exploration, design decisions are taken, the system is partitioned, and the specification for its components is detailed. In this phase, power budgets are given and power constraints are inferred. As soon as new components are designed (and synthesized), their power dissipation is evaluated and checked against the initial specification. Power estimation in this step must obey to much more stringent accuracy

requirements than those needed for early exploration, since absolute, rather than relative power figures must be determined. The implementation of the design is available in this phase, and the lack of specification is not an issue: accurate power estimates for gate or switch-level descriptions can then be obtained.

The literature on power estimation for design validation is extensive. Most techniques target the estimation of the *average power dissipation*, a scalar quantity that compactly represents the power consumption over a long period of time. According to Najm [2], gate and switch-level average power estimation methods belong to two main categories: *Probabilistic* (or *Static*) and *Statistical* (or *Dynamic*). The distinctive feature of probabilistic techniques is that they do not require the explicit generation of the input streams to be simulated; rather, they directly propagate through the circuit user-specified input probabilities and correlations to estimate the circuit's internal activity. Statistical techniques require to explicitly simulate typical input streams, which are usually randomly generated under some user-provided constraints concerning the switching probabilities of the circuit inputs. To obtain trustworthy power estimates, very long streams are needed; therefore, statistical methods are very time consuming.

In many practical cases, streams of patterns representing typical usage of the circuit are available (or can be produced with a relatively low effort by the designer). Such streams are usually very long; therefore, accurate power simulation is inherently slow and extremely expensive from the computational standpoint. For this reason, a lot of recent research has focussed on the development of techniques that limit the effort required to carry out an accurate estimation of the average power dissipated by a circuit.

We tackle the problem of reducing the simulation time from a novel point of view. It has been demonstrated in a number of cases [3]–[5] that average power does not fully characterize the power dissipation of real-life circuits. Different levels of consumption are in fact observed depending on the state of operation of the system. This is particularly true for systems that adopt dynamic power management [6], [7]. In these situations, a *time-dependent power waveform* that can track sudden variations of power consumption over time can clearly provide much more complete information. We thus propose a technique that can be used to quickly, yet accurately determine the power waveform, as well as the average power, for circuits whose power dissipation follows an “up–down staircase” curve.

We leverage multilevel simulation to extract high-level information that is used to select a subset (sample) of the entire input stream. Accurate power simulation is then performed on the selected subset. One important strength of our technique is that

Manuscript received February 5, 1998; revised November 24, 1999. This work was supported in part by the National Science Foundation (NSF) under Contract MIP-9421129 and in part by the Advanced Research Projects Agency (ARPA) under Contract DABT-63-95-C-0049. This paper was recommended by Associate Editor M. Pedram.

L. Benini is with the Università di Bologna, Dip. di Elettronica, Informatica e Sistemistica, Bologna 40136, Italy.

G. De Micheli is with Stanford University, Computer Systems Laboratory, Stanford, CA 94305 USA.

E. Macii, M. Poncino, and R. Scarsi are with the Politecnico di Torino, Dip. di Automatica e Informatica, Turin 10129, Italy (e-mail: enrico@athena.polito.it).

Publisher Item Identifier S 0278-0070(00)03225-5.

the selection of the sample can be performed *on-line*, while the high-level simulation is carried on.

From the user point of view, a fast cycle-based simulation is performed on the complete input stream. During simulation, our tool monitors an *indicator function* which provides information on the variations of the input stream and power dissipation. Accurate power simulation is automatically dispatched when needed for tracking the changes in the “local” mean value of the input stream. At the end of the cycle-based simulation, data on the time-varying power consumption of the system are available at the price of a modest time overhead (due to the accurate simulation performed on a small fraction of the stream). The tool automatically constructs the time-domain power waveform and computes its average.

The indicator function does not necessarily depend on the primary inputs only. As long as the internal state of the system is available during cycle-based simulation, its value can be taken into account as well. The same holds for the system outputs. In general, any information available during cycle-based simulation can be exploited for constructing an indicator function. Obviously, there is a tradeoff between the accuracy of such function and the overhead it imposes on cycle-based simulation. Our method is practical only if it guarantees that  $T_c + T_{\text{ind}} + T_{\text{sample}} \ll T_{\text{pow}}$ , where  $T_c$  is the time spent in fast cycle-based simulation,  $T_{\text{ind}}$  is the overhead for the computation of the indicator function,  $T_{\text{sample}}$  is the time required to perform accurate power simulation on the sample, and  $T_{\text{pow}}$  is the time required to perform accurate power simulation on the entire input stream.

Two recently proposed techniques have close relationship to ours. The power ratio method [8] is based on multilevel simulation. It assumes that the power estimate  $P_{\text{est}}$  provided by high-level fast power simulation is proportional to the power dissipation  $P_{\text{acc}}$  estimated with accurate simulation. In symbols:  $P_{\text{acc}} = P_{\text{est}} \cdot K$ . The unknown proportionality constant,  $K$ , is simply obtained by computing both  $P_{\text{acc}}$  and  $P_{\text{est}}$  for a small subset of the complete input stream, and by subsequently taking their ratio. Unfortunately, for deterministic, designer-supplied input streams,  $K$  is usually far from being constant; in these cases, the method is no longer applicable and may lead to inaccurate estimates.

Stratified random sampling [9], though developed as an alternative to Monte Carlo methods [10], has some similarities to the technique we propose in this paper. In fact, it generally requires only a small number of accurate simulations to achieve power estimates which are acceptable and, as for our method, it exploits a high-level power estimator to direct the choice of the patterns that will be simulated with high accuracy. In the form it has been presented, however, stratified random sampling targets the estimation of average power of input streams for which some statistical constraints, such as error and confidence level, are given; in addition, its applicability to the construction of time-dependent power waveforms has not yet been investigated. Finally, its effectiveness has been demonstrated only for combinational circuits. On the contrary, our approach works for purely deterministic input streams, it can provide the user with information on the average power values, as well as the changes of power dissipation over time, and it properly and accurately handles sequential circuits.

The tool has been benchmarked on example circuits taken from the Iscas’85 [11] and Iscas’89 [12] suites, as well as on a programmable digital filter. Given a circuit and an input stream, we have compared the average power value calculated using our technique to that obtained by switch-level simulation [13] of the entire stream. Results are satisfactory, the average error being below 2% for the combinational examples and below 6% for the sequential circuits. Moreover, the estimated power waveforms match closely the actual ones. Concerning the average simulation speed-up, it is around 50X for the combinational benchmarks and around 20X for the sequential examples.

The rest of the manuscript is organized as follows. Section II shows how the *up-down staircase average* behavior of a circuit can be tracked using an indicator function, and introduces a set of such functions that can be fruitfully used for constructing a power waveform. In Section III, we describe the multilevel simulation approach and we show how the indicator function is used to trigger the accurate power simulation engine. In Section IV, we present interpolation schemes for the extraction of the complete power waveform. Experimental results are provided in Section V. Section VI concludes the paper.

## II. TIME-DEPENDENT POWER WAVEFORMS

### A. Staircase Average Behavior

Consider a measurable property  $P(T)$  of a circuit that changes over time (in our case, power is the property of interest). We assume discretized time. A time quantum is called *cycle*. The property has an *up-down staircase average* behavior, *staircase* for brevity, over time when its running (or sliding) average taken over a window of  $N$  cycles,  $P_N(T)$ , changes in a nonmonotonic fashion, with long plateaus where  $P_N(T)$  is approximately constant, mixed with regions where it changes rapidly. Fig. 1(a) shows the diagram of a time-varying property  $P(T)$  with staircase behavior, and Fig. 1(b) shows the diagram of  $P_N(T)$ . The value of  $N$  (the averaging length) is shown as well. Notice that  $P_N(T)$  is not defined for the first  $N$  time points.

Extensive experimentation [14], [15] has shown that, when the input streams are randomly generated and composed of uniformly distributed, uncorrelated patterns, the power dissipation does not have a staircase behavior. On the other hand, the measured short-term average power dissipation of many real-life circuits has a staircase behavior [3]–[5], [7]. The key observation is that such behavior appears to be induced by the input streams which are processed by the circuits in typical operating conditions. This is certainly the case for power-manageable circuits, where the value of some inputs determines the mode of operation and, consequently, the power consumption level [6], [7].

Our target is to accurately estimate the power dissipation induced in a circuit by realistic input streams, which are not uniformly distributed, nor uncorrelated. This is a challenging task, because it requires a difficult tradeoff choice. The number of samples,  $N$ , taken to compute the running average is the key parameter for achieving the desired accuracy. If we choose an excessively large  $N$ , we loose accuracy in the estimation of the average power waveform  $P_N(T)$ . The larger  $N$ , the more  $P_N(T)$  resembles to a constant. On the other hand, if  $N$  is too short, the

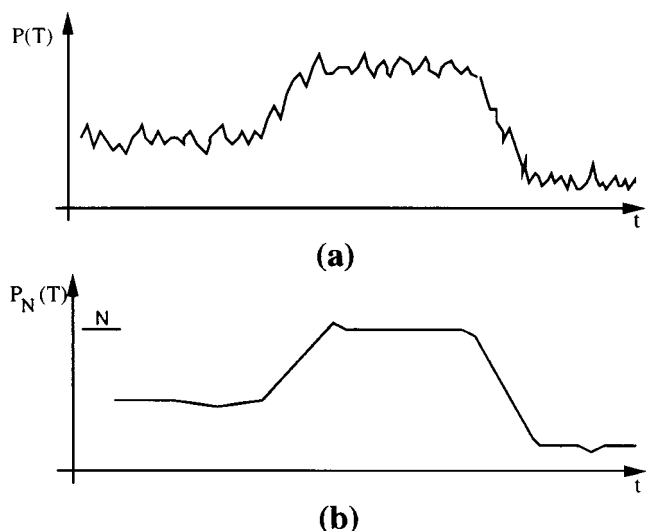


Fig. 1. Waveforms for (a)  $P(T)$  and (b)  $P_N(T)$ .

noise in the estimation of the average power will be too large (because of the effect of pattern dependency) and accuracy will be compromised as well.

Power estimation in the staircase situation is further complicated by computational issues. As explained in the introduction, we cannot afford to simulate entire input streams with an accurate power simulator. Therefore, we exploit information available during fast high-level simulation (hereafter called *Level 1* for brevity) to reduce the number of slow, accurate low-level simulations (hereafter called *Level 2*) needed to estimate the power. Differently from the existing techniques, this information is used to estimate how power changes over time when the given input stream is applied at the circuit inputs.

### B. The Indicator Function

We call *indicator function* the Level 1 information exploited to track the variations in switching activity of the input stream. The basic requirement for the indicator function is that it must change over time, and its changes should be related to the variations of the power dissipation. Notice that this is a milder requirement than proportionality (as requested by the power ratio method).

We propose alternative indicator functions based on sampling the (zero-delay) switching activity and the value of the nodes in Level 1 simulation. The sampling point selection process allows us to:

- tradeoff accuracy for computational overhead (increasing the number of sampling points increases the overhead in Level 1 simulation);
- exploit designer knowledge, if possible, to select specific nodes that should be sampled to increase the accuracy of the indicator.

We focus on indicator functions that do not require any designer intervention in the choice of the sampling points. Our target is to build a fully automated procedure for power estimation that can be applied successfully even without complete understanding of the design functionality and hardware architecture. In [16], it was observed that there exists strong correla-

tion between power dissipation in a combinational logic circuit and the switching activity at its inputs and outputs. A similar conclusion was reached in [17], [18], where input and output switching activities were represented by *entropy* values.

The basic claim in [16] is that power dissipation can be predicted with reasonable accuracy by computing a function of input/output (I/O) switching. In symbols

$$P^{n+1} = f(i_1^n \oplus i_1^{n+1}, \dots, i_{n_i}^n \oplus i_{n_i}^{n+1}, o_1^n \oplus o_1^{n+1}, \dots, o_{n_o}^n \oplus o_{n_o}^{n+1}) \quad (1)$$

where  $P^{n+1}$  is the power dissipated during cycle number  $n + 1$ ,  $i_k^n$  and  $i_k^{n+1}$  are, respectively, the values of input  $k$  at the beginning of cycles  $n$  and  $n + 1$ ,  $o_k^n$  and  $o_k^{n+1}$  are, respectively, the output values at the end of cycles  $n$  and  $n + 1$ . The function  $f$  is a general (possibly nonlinear) mapping  $f : B^{n_i+n_o} \rightarrow \mathcal{R}$ .

The choice of the sampling points for the indicator function is based on a similar assumption. We propose four different criteria, listed below in order of increasing accuracy and computational overhead.

- Sampling only the primary inputs. The main limitation of this criterion is that it does not consider any information on how input switching propagates through the logic. Moreover, for sequential circuits, the effect of internal state on power dissipation is not taken into account. On the other hand, this criterion has minimum overhead. Indeed, Level 1 simulation is not even needed, and analysis of the input stream is sufficient to extract the required information.
- Sampling primary inputs and outputs. This criterion increases the information on internal switching, since the end effect of the input propagation is sampled. Again, this criterion is targeted towards combinational circuits, since it does not account for internal state. Compared to the previous one, this choice of sampling points has higher overhead: the number of sampling points increases and, more importantly, Level 1 simulation must be executed for computing the correct output values.
- Sampling primary inputs and outputs, as well as the inputs and outputs of the flip-flops. This is the full cycle boundary information that is usually available in Level 1 simulation (which is cycle accurate). This criterion has higher overhead than the first two (the number of sampling points is greatly increased), but it is well suited for dealing with sequential circuits, because the internal state is exposed.
- Estimating the full zero-delay activity (as in the power ratio method and in stratified random sampling), including internal nodes in the combinational logic. This criterion has high overhead, because it prevents fast compiled cycle-based simulation (at Level 1) to “compile away” internal nodes. Moreover, it is not possible to perform Level 1 simulation at a level of abstraction higher than the logic level, because the values of the internal nodes cannot be sampled (they may not be instantiated). Although zero-delay simulation is still much faster than full-delay event-driven simulation (or switch-level simulation), it may be more than one order of magnitude slower than compiled, cycle-based simulation.

The computation of the indicator function requires not only the choice of the sampling points, but also the specification of how the sampled information should be used to compute a measure that tracks the temporal variations of the average power. The main problem is that power is strongly pattern dependent: its value can change widely in successive clock cycles. Moreover, we are not interested in tracking the pattern-by-pattern variations of power but, rather, the variations of the short-term average power. For these reasons, the indicator function must include a short-term time averaging operation. The simplest one is then the number of switching events averaged over a time window of size  $N_c$

$$I(T) = \frac{\sum_{t=T-N_c}^T \sum_{i \in S} \rho_i^t}{N_c} \quad (2)$$

where  $S$  is the set of sampling points,  $\rho_i^t$  are Boolean variables that have value 1 when the sampling point  $i$  switches between cycle  $t-1$  and  $t$  ( $\rho_i^t = i^{t-1} \oplus i^t$ ).

The choice of  $N_c$  in (2) is critical. It should be long enough to smooth out fluctuations due to single-pattern dependence, but it should be short enough to capture the staircase behavior. The issue of how to select  $N_c$  will be discussed in Section III.

Equation (2) defines the indicator as a function of time. The short-term average can be seen as a *sliding window*: at time  $T$ , we consider  $N_c$  sets of values of the sampling points, one for each simulation cycle from time  $T - N_c$  to  $T$ . The sliding window average is well suited to estimate variations over time, because it evolves in parallel with the advancement of the global simulation time. This is in sharp contrast with stratified random sampling, where samples can be selected and averaged together in any order, without any constraint on temporal adjacency.

The simple time-averaged sum of switching events of (2) can be replaced by more complex indicator functions. The one we use computes the weighted sum of switching events averaged over the window of size  $N_c$

$$I(T) = \frac{\sum_{t=T-N_c}^T \sum_{i \in S} w_i \rho_i^t}{N_c}. \quad (3)$$

In this case,  $\rho_i^t$  of (3) is replaced by  $w_i \rho_i^t$ . The weights  $w_i$  represent the relative influence of the switching of sampling point  $i$ , and can be user-specified. As a default, they are all set to the value 1; therefore, (3) reduces to the standard time-average sum of switching events.

### III. MULTILEVEL POWER SIMULATION

The multilevel simulation engine we propose is conceptually simple. Given a long stream of  $N_{\text{tot}}$  input patterns, together with a high-level and a low-level description of the circuit under analysis, the Level 1 simulation is started, whose tasks are the following:

- 1) monitoring the sampling points required for the computation of the indicator function;
- 2) computing the indicator function  $I(T)$ ;
- 3) performing the *staircase test* on  $I(T)$ .

The purpose of the staircase test is to decide, based on the changes over time of  $I(T)$ , when to fire Level 2 simulation. Whenever the staircase test triggers, the input and state values are extracted from Level 1 information and used to set the initial state for Level 2 simulation (this step is essential when dealing with sequential circuits). After that, the multilevel simulation can proceed in lock-step.

Level 2 simulation continues until a *stopping criterion* is satisfied. Such criterion decides when sufficient accurate power data have been collected to obtain a reliable short-term average power estimate  $P_{\text{avg}}(T)$  (notice that  $P_{\text{avg}}$  is a function of time). The value  $P_{\text{avg}}(T)$  is a point in the average power waveform.  $T$  is conventionally set to the value assumed in the last cycle of Level 2 simulation.

At the end of the simulation, all  $N_{\text{tot}}$  patterns have been simulated at Level 1, but only  $N_{\text{sample}}$  patterns have been simulated at Level 2. Typically,  $N_{\text{sample}} \ll N_{\text{tot}}$ . The end result is a set of values

$$\mathcal{P} = \{P_{\text{avg}}(T_1), \dots, P_{\text{avg}}(T_s)\} \quad (4)$$

where  $T_1 < \dots < T_s$ . The power values  $P_{\text{avg}}(T_i)$  and the times  $T_i$  are a set of samples of the average power *waveform* of the circuit.

The definition of the estimation strategy is completed by addressing the following open issues.

- How to measure the efficiency and accuracy of multilevel simulation.
- How to choose the length  $N_c$  of the sliding window average needed for computing  $I(T)$ .
- How to use the value of  $I(T)$  to decide when firing accurate simulation to track the staircase behavior (i.e., providing the definition of the staircase test).
- How long to run accurate power simulation every time it is started by the staircase test on  $I(T)$  (i.e., providing the definition of the stopping criterion).

#### A. Efficiency and Accuracy Metrics

The performance of the multilevel scheme is measured by the speed-up achieved with respect to accurate simulation of the entire input stream, because this is the only known way to extract complete power information of systems whose power holds a staircase behavior. The speed-up is defined by

$$\text{SP} = \frac{T_{\text{pow}}}{T_c + T_{\text{ind}} + T_{\text{sample}}} \quad (5)$$

where  $T_{\text{pow}}$  is the time needed to simulate the circuit for  $N_{\text{tot}}$  cycles at Level 2,  $T_c$  is the time for simulating  $N_{\text{tot}}$  cycles at Level 1,  $T_{\text{ind}}$  is the time overhead caused by the calculation of both the indicator function and the staircase tests, and  $T_{\text{sample}}$  is the time required to simulate  $N_{\text{sample}}$  cycles at Level 2 during multilevel simulation.

The accuracy of the multilevel scheme can be measured by comparing the average power,  $P_{\text{avg}}$ , obtained by simulating the complete input stream at Level 2 with the estimate,  $P_{\text{avg}}^{\text{est}}$ , obtained by averaging the power samples  $P_{\text{avg}}(T_i)$ . Clearly, the latter must be weighted with the *duration* of the samples:

$P_{\text{avg}}^{\text{est}} = \sum_{i=1}^s \tau_i P_{\text{avg}}(T_i)$ . The duration  $\tau_i$  of  $P_{\text{avg}}(T_i)$  is defined as

$$\tau_i = \frac{T_{i+1} - T_i}{N_{\text{tot}} T_{\text{clk}}} \quad (6)$$

where  $T_{\text{clk}}$  is the clock period of the simulator. The weighted average is needed because our technique extracts new power samples only when the staircase behavior is detected, i.e., when we are moving from one plateau to another. If the average power remains on a plateau for a long time, only one power sample is extracted, but it represents the average power dissipation of a very long time interval.

Although accurate average power estimation over the entire input stream is a minimum requirement, error on the average power estimation is not a reliable measure of the accuracy of our method in tracking the power waveform. Thus, we need to define a tighter error metric. Observed that  $P_{\text{avg}}^{\text{est}}(T)$  and  $P_{\text{avg}}(T)$  are both time-dependent functions (i.e., waveforms), a good metric for estimating the “similarity” of two functions over an interval is the *root-mean-square relative error* (RMSRE). In our discrete-time setting, RMSRE is defined as

$$\text{RMSRE} = \frac{1}{P_{\text{avg}}} \sqrt{\sum_{T \in [N_c, N_{\text{tot}}]} (P_{\text{avg}}^{\text{est}}(T) - P_{\text{avg}}(T))^2} \quad (7)$$

where  $P_{\text{avg}}$  without time dependency is the total average power. One difficulty in estimating RMSRE lies in the fact that our method computes  $P_{\text{avg}}^{\text{est}}(T)$  only in a small subset of the entire interval, namely the sampling points  $T_i$ . Hence, we need to define an *interpolation scheme* that, given the values  $P_{\text{avg}}^{\text{est}}(T_i)$ , extracts the values of  $P_{\text{avg}}^{\text{est}}$  for all remaining  $T \neq T_i$ . Interpolation schemes are discussed in Section IV.

### B. Choosing the Window Size $N_c$

The choice of the parameter  $N_c$ , that is, the length of the sliding window for the running average, is key. If  $N_c$  is too small, the value of  $I(T)$  will be noisy: the pattern-dependent fluctuations of the indicator function may change the value of  $I(T)$  too much and too rapidly. As a result, it may become impossible to discern slow variations of  $I(T)$  due to staircase behavior from fast variations due to pattern dependence. The consequence of this problem is that the staircase test is triggered too often and too many Level 2 simulations are executed, causing a marked slow-down of the multilevel simulation engine.

On the other hand, if  $N_c$  is too large,  $I(T)$  may have excessive *inertia* and change too slowly. In this case, the variations of  $I(T)$  over time may be smoothed down to a constant average value that does not represent the variation of the average value over time. As a consequence, the staircase test is almost never satisfied. The multilevel simulation becomes fast (a minimum number of Level 2 simulations is executed), but accuracy in tracking the power waveform is lost.

To avoid both pitfalls, the choice of  $N_c$  must be a compromise between good tracking capability and noisiness. Our procedure for choosing  $N_c$  is based on a calibration process. We move from the observation that, when uniformly distributed, uncorrelated patterns are fed to the system, the average power does not have

a staircase behavior and tends to converge to a constant value. We conjecture that the same holds for the indicator functions.

Before starting the multilevel simulation on the input stream, we perform a calibration Level 1 simulation with independent, uncorrelated input patterns. The simulation is run until convergence is reached on the average value of  $I$ . Assume that the calibration simulation converges in  $N_{M_I}$  cycles.  $N_{M_I}$  is the number of simulation cycles needed for convergence on a stable average value of  $I$  when the input patterns are independent and uncorrelated. This value is significant for our purposes, because it gives us information on the number of cycles needed to smooth out only the power variations due to strong pattern dependence. The calibration simulation allows us to separate the effects. During calibration, the power does not have a staircase behavior, but it is still strongly pattern dependent.

The number of cycles for short-term average,  $N_c$ , is chosen to be

$$N_c = k N_{M_I} \quad (8)$$

where  $0 < k \leq 1$ . When  $k = 1$ , we have a *high inertia*  $I(T)$ . When  $k < 1$ , we may be able to track fast step-wise variations, but we may sometimes take a simple fluctuation due to strong pattern dependency at the beginning (the end) of a step. The choice depends on the target of the power estimation process. If we are mainly interested in obtaining a single average power value with maximum efficiency,  $k$  should be chosen close to one. On the contrary, if we need great accuracy on the estimate of the average power value, or we want to finely sample the power waveform,  $k$  should be close to 0.5. The price paid for the increase in accuracy is a longer runtime (a larger number of Level 2 simulations are required). Values of  $k < 0.5$  are not advisable for performance reasons.

### C. Staircase Test

During Level 1 simulation,  $I(T)$  is computed on every simulation cycle. Although  $I(T)$  is defined at every clock cycle, it is meaningless formulating a staircase test that involves decisions taken on time intervals shorter than  $N_c$ , being this the maximum resolution.

The test is based on an upper bound on variations of  $I(T)$ , and it is applied using the following procedure.

- Initially, Level 2 simulation is started to obtain the first power sample. Concurrently, Level 1 simulation is run and  $I(T)$  is computed. When Level 2 simulation is stopped (through the criterion described later), the value of  $I(T_0)$  at the stopping time  $T_0$  is stored.
- Level 1 simulation and the computation of the indicator function are carried on for  $T > T_0$ . The staircase test is disabled for a short *recovery interval*, that is, until  $T = T + \beta T_{\text{clock}} N_c$ , where  $\beta \geq 1$ . As mentioned above, the rationale for this choice is to provide a lower bound on the granularity at which the staircase behavior is tested.
- After the recovery interval, the staircase test is applied at every clock cycle  $T$ . A staircase behavior is detected when

$$\Delta|I(T)| > \gamma \Delta_{\text{max}}|I| \quad (9)$$

where  $\Delta|I(T)| = |I(T_0) - I(T)|$ ,  $\gamma$  is a coefficient smaller than one, and  $\Delta_{\max}|I|$  is an upper bound to variations of  $|I|$ . In the case of the simple indicator function of (2),  $\Delta_{\max}|I| = |S|$ , where  $|S|$  is the cardinality of the set of nodes in the circuit whose switching activity is sampled by the indicator function.

- If the test is satisfied, Level 2 simulation is triggered. Upon stopping of the latter (at time  $T_2$ ),  $I(T_2)$  is stored and the process is restarted.

Parameter  $\gamma$  controls the sensitivity of the test; its typical range is  $0.1 < \gamma < 0.5$ . The choice of  $\gamma$  is again dictated by a tradeoff between sensitivity and accuracy in the estimation.

#### D. Stopping Criterion

The last decision to be taken is when to stop the Level 2 simulation after it is started by the staircase test. The simplest choice is to execute Level 2 simulation for a fixed number of clock cycles. Unfortunately, no simple criterion is available for the choice of such number.

We adopt an adaptive strategy, where the number of Level 2 simulations is variable and depends on the operating conditions. More specifically, the stopping criterion is based on a convergence test on the average power value. Although this choice is intuitively attractive, it is heuristic and relies on the assumption that on the short term, the average power dissipation appears nonstaircase, hence it converges rather rapidly to a constant average value.

Clearly, this assumption is not true in general. More in detail, two problems may arise. The first is premature convergence: Level 2 simulation is stopped because convergence on the average is reached too rapidly (for example, if, by chance, the power dissipated at the first two or three cycles of Level 2 simulation is almost constant). The second, and more serious, is lack of convergence. In this case, Level 2 simulation never stops. To mitigate both problems, we provide the following lower bound on the number of vectors that must be simulated in Level 2:  $\eta_L N_c$ , with  $\eta_L < 1$ , typically between 0.5 and 1. Similarly, the upper bound is  $\eta_U N_c$ , with  $\eta_U > 1$ , typically between 2 and 3. The rationale of these bounds is based on the observation that the convergence of power estimation is related to convergence of the indicator function, although it is expected to be slower.

#### IV. INTERPOLATION SCHEMES

The multilevel simulation engine of Section III computes a set of samples  $P_{\text{avg}}(T_i)$  of the average power waveform. The estimation of  $P_{\text{avg}}(T)$  for  $T \neq T_i$  is called *interpolation*. The simplest interpolation strategy is *piecewise constant*: the value of  $P_{\text{avg}}(T)$  is assumed to be equal to the last sampled value

$$P_{\text{avg}}^{\text{est}}(T) = P_{\text{avg}}(T_i) \quad T_i \leq T < T_{i+1}. \quad (10)$$

The piecewise constant interpolation is consistent with the assumption that average power has a staircase behavior, but has two main limitations. First, it performs quite poorly when the transitions between plateaus are smooth (or when the waveform is not constant nor fully staircase); second, it does not completely exploit the information provided by the indicator function.

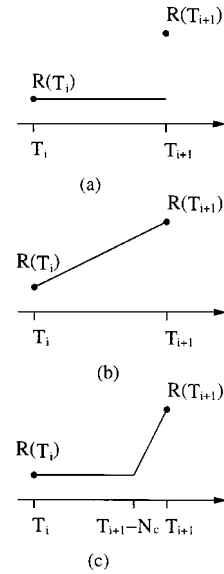


Fig. 2. (a) Discontinuous variation of  $R$ . (b) Linear smoothing of  $R$ . (c) Mixed smoothing of  $R$ .

We propose an enhanced interpolation scheme that exploits the availability of  $I(T)$  to improve accuracy. The method is based on the assumption that, locally,  $I(T)$  is proportional to  $P_{\text{avg}}(T)$ . In other words, once a sample  $P_{\text{avg}}(T_i)$  has been obtained, we can compute a proportionality constant  $R(T_i) = P_{\text{avg}}(T_i)/I(T_i)$ . Notice that the definition of  $R(T_i)$  is the same as the *power-ratio*. However, unlike [8], we do not assume that  $R$  is a constant across the simulation.

In our interpolation strategy,  $R(T_i)$  is computed at each power sample; then, its value is used to estimate  $P_{\text{avg}}(T)$  when  $T \neq T_i$ . In symbols

$$P_{\text{avg}}^{\text{est}}(T) = R(T_i)I(T) \quad T_i \leq T < T_{i+1}. \quad (11)$$

Experimentally we noticed that, although this scheme leads to much better predictions than the piecewise constant one, it still has limited accuracy because of the discontinuity on the values of  $R(T_i)$ . The difference between  $R(T_i)$  and  $R(T_{i+1})$  can be big, and this leads to increased estimation errors in the left neighborhood of the sampling points  $T_i$  (i.e., for cycles immediately before a sampling point).

Error can be further reduced by imposing a smooth transition between  $R(T_i)$  and  $R(T_{i+1})$ . We employ simple linear smoothing. Fig. 2(a) shows a discontinuity between two successive samples of  $R$ . Fig. 2(b) shows the simplest linear smoothing of  $R$ . Linear smoothing performs well for waveforms with short plateaus and numerous sampling points. The main limitation of linear smoothing is that it has limited accuracy for power waveforms with long plateaus and few (and well apart) sampling points. The accuracy loss is caused by the fact that when  $T_i$  and  $T_{i+1}$  are far apart, the value of  $R$  remains close to  $R(T_i)$  for a long time before starting to change. A more robust and accurate smoothing is shown in Fig. 2(c).  $R(T)$  is held at the constant value  $R(T_i)$ , then it transitions linearly to  $R(T_{i+1})$ . The transition takes  $N_c$  cycles. This scheme gives good results for long plateaus and reduces the error around sampling points. The choice of  $N_c$  for the duration of the transition is dictated by

the fact that  $N_c$  is the “inertia” of the algorithm. In other words, transitions faster than  $N_c$  are not significant for power estimates averaged over  $N_c$ . Assuming that the entire variation of  $R(T)$  takes place in  $N_c$  cycles is consistent with the assumption of staircase power waveform, with steep edges, where the speed of variation is controlled by the window amplitude  $N_c$ .

More complex interpolation schemes could be adopted to further improve accuracy. In particular, full-fledged regression models (e.g., [19]) could be constructed. This would increase the complexity of the method and decrease its performance.

## V. EXPERIMENTAL RESULTS

In order to investigate the applicability and the accuracy of the proposed methodology, we have performed three sets of power estimation experiments, and we have checked the so computed results against the ones we have obtained through exhaustive transistor-level simulation using Irsim [13]. In the following sections, we report on our findings. Before that, we present an analysis on how the choice of some of the tuning parameters that can be selected by the user may impact the performance and accuracy of the method.

### A. Sensitivity Analysis

There are five user-selectable parameters that control the operations of the multilevel simulation algorithm described in Section III.

- $k$ , that controls the size of the sliding window  $N_c$ . Parameter  $k$  lies within the interval  $[0, 1]$ . Remember that, as  $k \rightarrow 1$ , the window size  $N_c \rightarrow N_{M_T}$ .
- $\beta$ , that defines the amplitude of the recovery interval, i.e., the number of clock cycles for which the staircase test is disabled after a Level 2 simulation is stopped. Parameter  $\beta$  lies within the interval  $[1, \infty]$ .
- $\gamma$ , that controls the sensitivity of the staircase test. Smaller values of  $\gamma$  imply higher sensitivity. Parameter  $\gamma$  lies within the interval  $[0, 1]$ .
- $\eta_L$  and  $\eta_U$ , that control, respectively, the minimum and the maximum number of Level 2 simulations to be performed when a power sample is taken. Parameter  $\eta_L$  lies within the interval  $[0, 1]$ , while parameter  $\eta_U$  lies within the interval  $[1, \infty]$ .

We study the sensitivity of the algorithm to the values such parameters can take on. Intuitively,  $k$  and  $\gamma$  are critical for tuning the accuracy and speed of the simulation, since  $k$  controls the short-term averaging of the power samples and  $\gamma$  controls the dispatching of Level 2 simulation. On the other hand,  $\beta$ ,  $\eta_L$  and  $\eta_U$  are not critical, because they control secondary features of the simulation engine. Experiments confirm this intuition: accuracy and simulation speed depend very weakly on  $\beta$ ,  $\eta_L$  and  $\eta_U$ . Hence, we have assigned them fixed values ( $\beta = 1$ ,  $\eta_L = 0.5$  and  $\eta_U = 2$ ) and we have not analyzed them further (these values are also used for the experiments in Sections V-B, V-C, and V-D). Conversely, we have studied the impact of parameters  $k$  and  $\gamma$  on efficiency and accuracy.

Fig. 3(a) and (b) shows the simulation accuracy (normalized RMSRE) and the number of Level 2 simulations as a function of  $k$  for a fixed value of  $\gamma$  ( $\gamma = 0.2$ ), respectively.

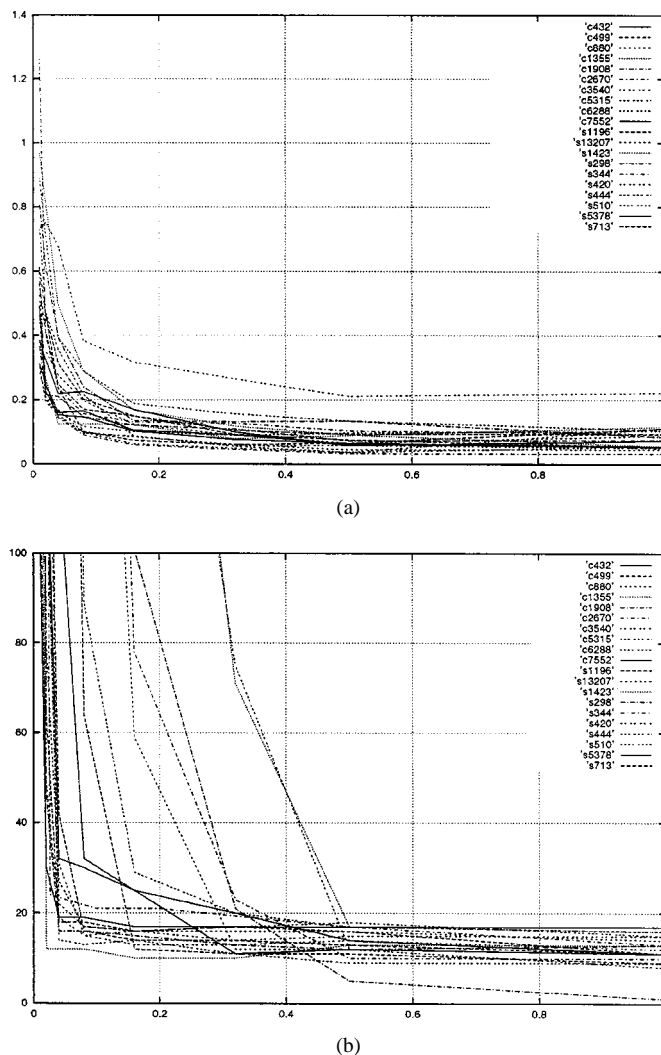


Fig. 3. (a) Normalized RMSRE. (b) Number of level 2 simulations versus  $k$ .

As the value of  $k$  gets smaller, the sliding window width decreases, and the value of  $I(T)$  becomes noisier. The macroscopic effect of the noisiness of  $I(T)$  is that it has larger variations on a short time scale with the result that the RMRSE increases, and Level 2 simulation is started much more often. Concerning the estimation accuracy [Fig. 3(a)], we can notice that it is almost constant over the full range of  $k$ , except for very small values, where it increases sensibly. Apparently then, the larger the value of  $k$ , the smaller the error. However, if we allow  $k$  to get values outside the range  $[0, 1]$  (typically  $k > 10$ ), the RMSRE starts increasing, because for large sliding windows the indicator function tends to lose its tracking capability, and it is not able to correctly predict large power variations.

Based on these considerations, the selection of  $k$  might still appear not critical, as long as average power is locally almost constant. However, if the target is to construct a power waveform, we would like to keep the tracking capability of the tool as high as possible, that is, to use smaller values of  $k$ . Therefore, we pick the smallest value of  $k$  that allows to achieve the required accuracy. The choice of  $k$  is clearly also affected by its impact on simulation time. As shown in Fig. 3(b), the number of Level 2 simulations increases as  $k$  decreases, whereas for small

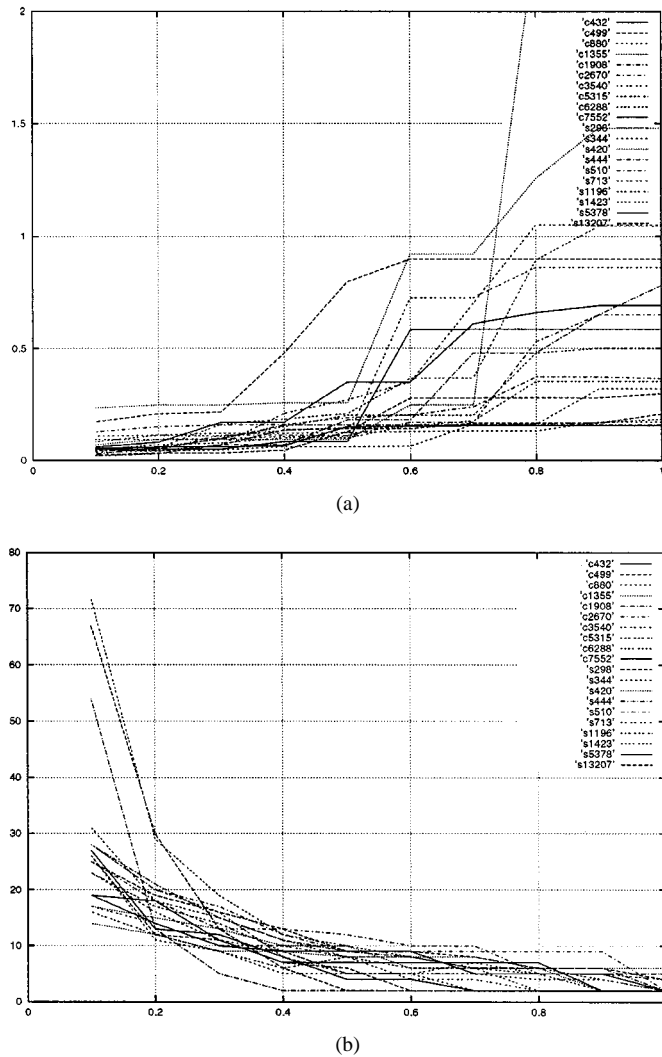


Fig. 4. (a) Normalized RMSRE and (b) number of level 2 simulations versus  $\gamma$ .

values of  $k$  ( $k < 0.5$ ) it increases dramatically. The shape of the curves gives hints on the minimum value of  $k$  that can be chosen. Putting together the constraints resulting from Fig. 3, we used the value  $k = 0.5$  for all the experiments of Sections V-B, V-C, and V-D. Fig. 4(a) and (b) shows the dependency of simulation accuracy (normalized RMSRE) and the number of Level 2 simulations on the value of parameter  $\gamma$  (for a fixed value of  $k = 0.5$ ), respectively.

As  $\gamma$  gets larger, the estimation error increases because the simulation engine cannot effectively track the staircase behavior of the power waveform. At the same time, for larger values of  $\gamma$ , the staircase test becomes less sensitive and Level 2 simulation is started only for very large variations of  $I(T)$ . The limiting case is when Level 2 simulation is performed only once at the beginning of the input stream. This happens when  $\gamma \geq 0.8$ . If  $\gamma$  is too small, accuracy increases but performance decrease, because Level 2 simulation is triggered too many times. Hence, the choice of the optimal value of  $\gamma$  is given by the tradeoff between accuracy and number of Level 2 simulations. Consistently with the two plots, we have set  $\gamma = 0.2$  for all the experiments described in Sections V-B, V-C, and V-D.

## B. Combinational Circuits

We have considered all the ISCAS'85 examples [11]. The circuits were optimized and mapped for area using SIS [20] onto a gate-library consisting of 2-input NAND and NOR gates, inverters, and buffers. Tables I and II collect the results of the experiments. Columns  $I$ ,  $O$ , and  $G$  of Table I give the number of circuit inputs, outputs, and gates, respectively. Column  $P_{avg}$  provides the actual values of average power obtained through complete transistor-level simulation of the input stream. Finally, column *Estimated Power* shows the values of average power obtained with the technique of this paper. The column is further split in three sets of columns, one for each of the three sampling points selection criteria we have considered, namely, input (*In*), input/output (*In-Out*), and global switching activity (*Intern*). Each column shows the estimated average power (column  $P_{avg}^{est}$ ), the absolute value of the relative error, with respect to the actual average power (column  $|E|$ ), and the value of the RMSRE. For the same benchmarks, Table II gives the time required to estimate the average power through complete transistor-level simulation (column  $T_{pow}$ ), and for each of the three indicator functions, gives the time  $T_c + T_{ind} + T_{sample}$  required to complete the estimation (column  $T$ ), the achieved speed-up, SP, and the number of Level 2 simulations triggered (column  $L2$ ). The size of the sliding window  $N_c$ , specific for each circuit, that we have used in the indicator functions is also shown in the right-most column.

All power values in Tables I and II, and in those included in the following sections, are expressed in milliwatts, while the CPU times are measured in seconds on a DEC AXP 1000/400 with 256 MB of memory.

Since the benchmark circuits do not come with typical input patterns, we built input streams trying to emulate real-life usage sequences. The generated input streams have very high temporal and spatial correlation (i.e., correlation between successive patterns and between input variables) and, more importantly, are highly nonstationary. The average input activity is changed abruptly and wide variations are imposed several times in the stream. Between variations, the average input activity is roughly constant, although the vectors are correlated. The complete streams consisted of 50 000 patterns.

The error between the average power values provided by the tool and the ones obtained through complete transistor-level simulation of the input stream are limited, namely, 1.715%, 1.117%, and 0.572%, depending on the selected sampling criterion. Also, the RMSRE is low (0.071, 0.062, and 0.049). On the other hand, the simulation speed-up is substantial (50.96X, 49.08X, and 47.13X).

As expected, a more accurate sampling criterion yields increasingly accurate estimates. However, it should be observed that the difference in accuracy is quite small. This is of interest when Level 1 simulation is carried out at a high level of abstraction, e.g., behavioral simulation. In this case, the knowledge of the internal switchings is not available, and the designer can rely only on I/O information.

Fig. 5 shows, for benchmark *c432*, how the three sampling criteria (the top three plots, from left to right), behave in tracking the power waveform obtained by simulation of the given stream.



TABLE I  
 ESTIMATION RESULTS FOR COMBINATIONAL CIRCUITS: AVERAGE POWER, ERROR, AND RMSRE

Circuit	I	O	G	$P_{avg}$	Estimated Power								
					In			In-Out			Intern		
					$P_{avg}^{est}$	E	RMSRE	$P_{avg}^{est}$	E	RMSRE	$P_{avg}^{est}$	E	RMSRE
c432	36	6	265	0.667	0.684	2.548%	0.084	0.680	1.949%	0.075	0.656	1.649%	0.060
c499	41	32	525	1.344	1.334	0.744%	0.051	1.346	0.148%	0.050	1.346	0.148%	0.049
c880	60	26	431	1.246	1.263	1.364%	0.036	1.258	0.963%	0.034	1.252	0.481%	0.031
c1355	41	32	525	1.323	1.252	5.366%	0.078	1.355	2.418%	0.076	1.316	0.529%	0.023
c1908	33	25	529	1.726	1.713	0.753%	0.052	1.715	0.637%	0.045	1.730	0.231%	0.037
c2670	233	140	808	2.660	2.653	0.263%	0.064	2.661	0.037%	0.062	2.661	0.037%	0.056
c3540	50	22	1289	5.207	5.279	1.382%	0.063	5.262	1.056%	0.058	5.182	0.480%	0.058
c5315	178	123	1759	6.397	6.417	0.312%	0.056	6.410	0.203%	0.039	6.397	0.000%	0.034
c6288	32	32	3240	10.050	10.456	4.039%	0.132	10.200	1.492%	0.096	10.080	0.298%	0.065
c7552	207	108	2314	10.366	10.670	2.932%	0.092	10.602	2.276%	0.082	10.560	1.871%	0.078
<b>Average</b>													
						1.715%	0.071		1.117%	0.062		0.572%	0.049

 TABLE II  
 ESTIMATION RESULTS FOR COMBINATIONAL CIRCUITS: TIME AND SPEED-UP

Circuit	$T_{pow}$	In			In-Out			Intern			$N_c$
		T	SP	L2	T	SP	L2	T	SP	L2	
c432	734	10	73.4	14	10	73.4	14	10	70.8	14	153
c499	697	10	69.7	15	10	69.7	15	11	63.2	15	117
c880	531	12	44.3	16	12	44.3	16	12	43.6	18	84
c1355	1007	9	111.9	8	11	92.6	12	10	101.6	10	234
c1908	659	19	34.7	17	18	36.6	15	23	28.1	20	191
c2670	1101	21	52.4	17	21	52.4	17	25	44.6	16	119
c3540	9675	258	37.5	16	250	38.7	16	229	42.2	14	317
c5315	6063	180	33.7	15	189	32.1	16	205	29.5	18	309
c6288	58410	3123	18.7	9	3123	18.7	9	3390	17.2	9	296
c7552	50325	1512	33.3	14	1558	32.3	14	1647	30.5	14	419
<b>Average</b>			50.96			49.08			47.13		

The waveform, obtained with the indicator function based on I/O sampling and the mixed smoothing interpolation method, is the bottom left plot. The actual power waveform is the bottom right plot. Obviously, the scales for the ordinates of the top three plots are different from those of the bottom ones, since the indicator functions provide numbers of weighted switching events, as opposed to actual power values. Comparing the two bottom plots, we observe that the indicator function based on I/O activity provides sufficient information to track the staircase behavior of the power for combinational circuits.

To check the robustness of the method, we applied it to input streams that do not cause a staircase behavior in the power dissipated by the circuit. Table III shows the results obtained using the I/O sampling points selection criterion. Column *Estimated Power* provides the various data and it is further split in two sets of columns: the first refers to a uniform random input stream, the second to a stream that presents slow variations (sinusoid) of the switching activity with respect to time. As expected, the RMSRE for the *Random* stream is very low and the number of Level 2 simulations is exactly 1. In fact, since this type of stream does not present any wide variation of the switching activity, the method requires only one low-level simulation at the beginning of the simulation. Also, the RMSRE for the *Sinusoidal* streams is quite low, indicating the effectiveness of the two-level simu-

lator in tracking the behavior of streams with different activity profiles. Clearly, the number of Level 2 simulations is high, due to the continuous variations of the switching activity.

### C. Sequential Circuits

We have selected a subset of the ISCAS'89 benchmarks [12], so as to create a representative sample, in terms of functionality, size, and topological structure (sequential depth, number of inputs, outputs, and flip-flops), of the existing examples. Tables IV and V show the results obtained for these circuits (column FF indicates the number of flip-flops). For the experiments we have used the following indicator functions: 1) primary input and primary output activity (*In-Out*); 2) primary input, primary output and state activity (*In-Out-State*); 3) complete internal activity (*Intern*).

As for the combinational examples, estimates are accurate, concerning both average power (5.703%, 3.823%, and 2.246%, depending on the sampling points selection approach) and RMSRE (0.135, 0.116, and 0.082). On the other hand, the speed-up with respect to exhaustive simulation is not as high as in the case of combinational circuits (21.63X, 20.28X, and 17.73X); this is caused by the fact that the convergence of each Level 2 simulation is usually slower, due to the presence of the circuit state.

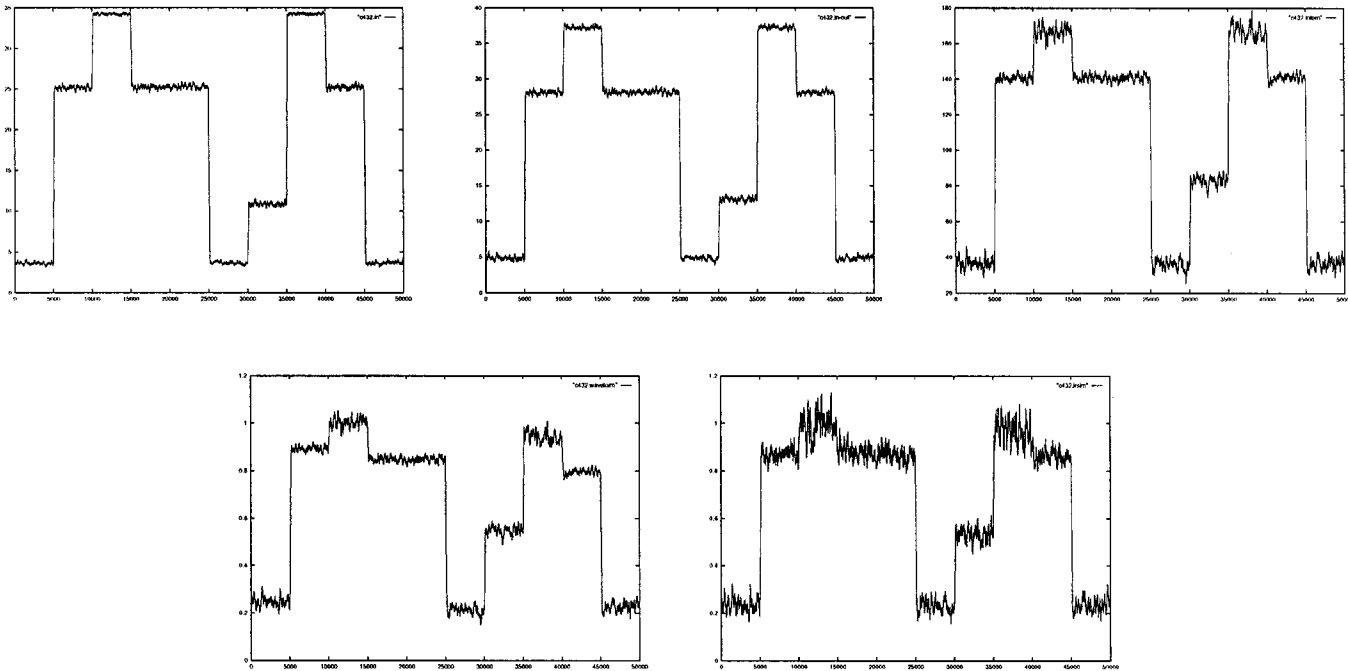


Fig. 5. Indicator Functions, mixed smoothing interpolation, and power profile for circuit c432.

TABLE III  
ESTIMATION RESULTS FOR COMBINATIONAL CIRCUITS: RANDOM AND SINUSOIDAL STREAMS

Circuit	Estimated Power											
	Random Stream						Sinusoidal Stream					
	$P_{avg}$	$P_{avg}^{est}$	$ E \%$	RMSRE	SP	$L_2$	$P_{avg}$	$P_{avg}^{est}$	$ E \%$	RMSRE	SP	$L_2$
c432	0.736	0.723	1.766%	0.078	294.12	1	0.652	0.652	0%	0.078	1.68	117
c499	1.63	1.62	0.613%	0.016	326.48	1	1.402	1.407	0.356%	0.021	1.71	135
c880	1.245	1.237	0.642%	0.031	272.51	1	1.113	1.146	2.964%	0.077	1.14	150
c1355	1.633	1.628	0.306%	0.023	227.13	1	1.475	1.484	0.610%	0.095	1.12	153
c1908	1.926	1.961	1.817%	0.033	252.87	1	1.624	1.652	1.724%	0.088	1.42	132
c2670	3.131	3.129	0.063%	0.014	232.25	1	2.813	2.861	1.706%	0.061	1.81	101
c3540	5.156	4.961	3.782%	0.047	290.52	1	4.572	4.570	0.043%	0.063	1.62	107
c5315	7.949	8.011	0.779%	0.012	304.15	1	6.708	6.828	1.789%	0.074	1.53	118
c6288	12.287	12.831	4.427%	0.047	234.36	1	10.557	10.777	2.083%	0.087	1.84	96
c7552	13.649	13.713	0.468%	0.023	253.76	1	12.253	12.346	0.759%	0.052	1.87	97
<b>Average</b>			1.466%	0.032	268.81	1			1.203%	0.069	1.57	120.6

TABLE IV  
ESTIMATION RESULTS FOR SEQUENTIAL CIRCUITS: AVERAGE POWER, ERROR, AND RMSRE

Circuit	I	O	FF	G	$P_{avg}$	Estimated Power								
						In-Out			In-Out-State			Intern		
						$P_{avg}^{est}$	$ E \%$	RMSRE	$P_{avg}^{est}$	$ E \%$	RMSRE	$P_{avg}^{est}$	$ E \%$	RMSRE
s298	3	6	14	136	0.050	0.048	4.000%	0.066	0.049	2.000%	0.040	0.050	0.000%	0.022
s344	9	11	15	151	0.082	0.080	2.439%	0.130	0.080	2.439%	0.112	0.082	0.000%	0.067
s420	18	1	16	168	0.041	0.034	17.073%	0.254	0.035	14.634%	0.254	0.036	12.195%	0.251
s444	3	6	21	181	0.046	0.048	4.347%	0.083	0.047	2.173%	0.058	0.046	0.000%	0.045
s510	19	7	6	268	0.096	0.087	9.375%	0.185	0.089	7.291%	0.149	0.092	4.166%	0.077
s713	35	23	17	189	0.059	0.060	1.694%	0.093	0.060	1.694%	0.094	0.060	1.694%	0.058
s1196	14	14	18	647	0.182	0.180	1.098%	0.083	0.180	1.098%	0.069	0.181	0.549%	0.065
s1423	17	5	74	757	0.255	0.235	7.842%	0.153	0.250	1.960%	0.111	0.253	0.784%	0.054
s5378	35	49	164	1585	0.446	0.436	2.242%	0.076	0.445	0.224%	0.065	0.445	0.224%	0.059
s13207	31	121	669	2693	0.491	0.525	6.924%	0.224	0.524	6.720%	0.208	0.505	2.851%	0.118
<b>Average</b>							5.703%	0.135		3.823%	0.116		2.246%	0.082

For some sequential circuit topologies, the temporal behavior of the indicator function based solely on I/O switching may not be informative enough about the actual power. This is because,

for circuits having far more latches than primary inputs, the degree of switching occurring at the primary inputs has a limited correlation to the overall activity (and thus power); rather, this is

TABLE V  
 ESTIMATION RESULTS FOR SEQUENTIAL CIRCUITS: TIME AND SPEED-UP

<i>Circuit</i>	$T_{pow}$	<i>In-Out</i>			<i>In-Out-State</i>			<i>Intern</i>			$N_c$
		<i>T</i>	<i>SP</i>	<i>L2</i>	<i>T</i>	<i>SP</i>	<i>L2</i>	<i>T</i>	<i>SP</i>	<i>L2</i>	
s298	442	30	14.7	31	31	14.4	31	31	14.4	31	120
s344	592	24	24.7	12	24	24.7	12	25	24.0	13	129
s420	489	22	22.2	15	22	22.8	14	26	18.5	16	117
s444	401	19	21.1	17	25	16.2	27	46	8.7	41	131
s510	685	37	18.5	16	39	17.4	17	56	12.3	24	121
s713	408	27	15.1	13	28	14.6	14	28	14.5	14	152
s1196	1554	63	24.7	12	74	21.1	14	68	22.8	13	125
s1423	2333	79	29.5	13	106	22.0	18	159	14.6	32	171
s5378	3912	185	21.1	14	172	22.8	13	190	20.6	15	162
s13207	2270	92	24.7	12	84	26.9	11	84	26.9	11	168
<b>Average</b>			<b>21.63</b>			<b>20.28</b>			<b>17.73</b>		

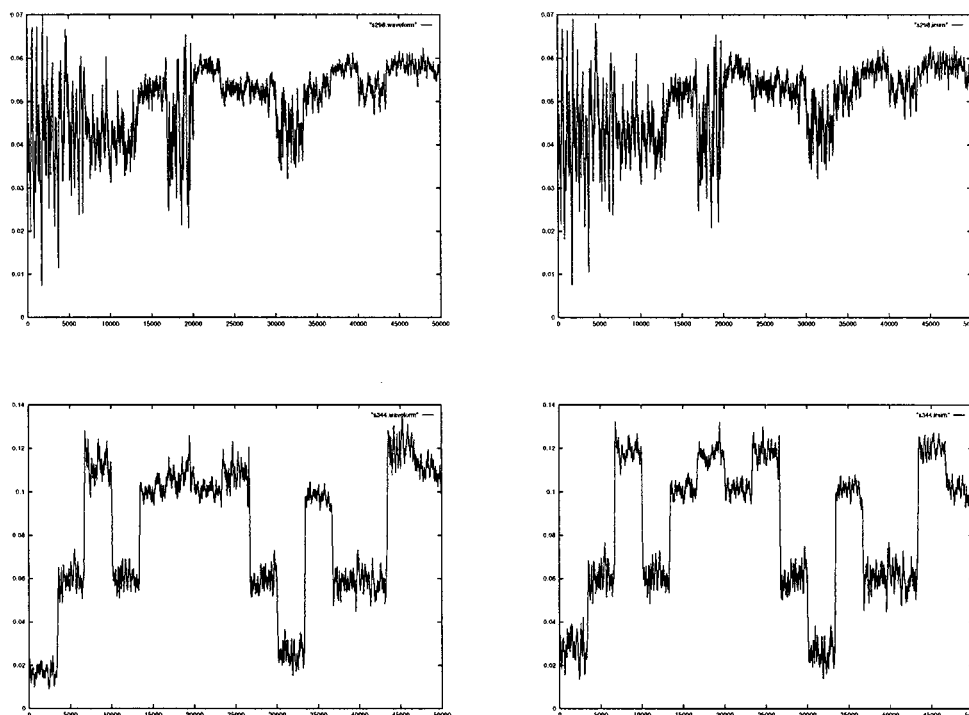


Fig. 6. Different behavior of two sequential benchmarks.

mainly driven by the switching at the present state lines, which induces most of the switching inside the combinational logic.

Fig. 6 shows a graphical view of this fact. The plots at the top of the figure refer to circuit s298. This circuit falls into the class of circuits just described, since it has 3 inputs and 14 latches. Although the input stream for the circuit has an average switching activity with staircase behavior, the power dissipation (the top right plot) has very weak correlation with the input switching activity (the staircase behavior is heavily smoothed out). Nevertheless, the behavior of power dissipation can be tracked, because the indicator function also observes the switching activity of the state variables. The mixed smoothing interpolation of the power dissipation is shown in the top left plot.

The plots at the bottom of the figure, on the other hand, refer to circuit s344; in this case, waveforms seem to exhibit a “combinational” behavior, in the sense that the input activity has a

considerable control over the internal degree of switching (and, consequently, power dissipation).

In general, we expect realistic sequential circuits to have a behavior similar to s298. In other words, sampling state information is key for achieving good accuracy. This conjecture is confirmed by the average power dissipation estimates. The accuracy of the estimate based on input, output and state activity is much better than that based on just input and output activity. Obviously, the most accurate estimate is obtained if we use the complete internal switching activity information (but, in this case, the computation of  $I(T)$  may be computationally expensive, because it requires gate-level zero-delay simulation).

Similarly to the case of combinational circuits, we show in Table VI some results for input streams that do not exhibit a staircase behavior in the power dissipated by the circuit. As for the combinational examples, the results for the *Random* streams

TABLE VI  
ESTIMATION RESULTS FOR SEQUENTIAL CIRCUITS: RANDOM AND SINUSOIDAL STREAMS

Circuit	Estimated Power											
	Random Stream						Sinusoidal Stream					
	$P_{avg}$	$P_{avg}^{est}$	$ E \%$	RMSRE	SP	$L_2$	$P_{avg}$	$P_{avg}^{est}$	$ E \%$	RMSRE	SP	$L_2$
s298	0.048	0.046	4.166%	0.048	226.71	1	0.049	0.047	4.082%	0.071	2.12	93
s344	0.084	0.087	3.571%	0.059	282.13	1	0.076	0.076	0%	0.091	1.78	97
s420	0.043	0.037	13.953%	0.151	191.06	1	0.042	0.041	2.381%	0.154	1.54	119
s444	0.044	0.043	2.272%	0.045	228.17	1	0.045	0.041	8.889%	0.118	2.46	75
s510	0.110	0.107	2.727%	0.057	244.84	1	0.098	0.095	3.061%	0.094	1.58	113
s713	0.085	0.088	3.529%	0.049	271.36	1	0.079	0.080	1.266%	0.033	1.37	127
s1196	0.337	0.330	2.077%	0.037	198.25	1	0.304	0.310	1.974%	0.044	1.91	93
s1423	0.310	0.301	2.903%	0.074	178.83	1	0.253	0.270	6.719%	0.161	1.86	106
s5378	0.591	0.605	2.368%	0.054	210.81	1	0.528	0.526	0.379%	0.037	2.09	98
s13207	0.465	0.455	2.150%	0.044	231.49	1	0.453	0.456	0.662%	0.108	1.77	103
<b>Average</b>			<b>3.971%</b>	<b>0.061</b>	<b>226.36</b>	<b>1</b>			<b>2.941%</b>	<b>0.091</b>	<b>1.85</b>	<b>102.4</b>

are good, whereas for the *Sinusoidal* streams accuracy comes at the price of an increased number of Level 2 simulations.

#### D. Case Study

The effectiveness of the power estimation methodology of this paper is best illustrated by analyzing its application to a real-life system. We designed a fully functional programmable digital filter. Starting from a behavioral description in Verilog, we synthesized a gate-level implementation using Synopsys Design Compiler, then we obtained the transistor-level implementation. The design contained 2190 gates (approximately 4000 transistors).

The flow graph of the filter is shown in Fig. 7(a). All coefficients are programmable, hence any transfer function with three forward and two backward coefficients can be implemented. The input, output, and coefficients are 8 bit wide. The high-level architecture of the design is shown in Fig. 7(b). The inputs are: IN (the input bus, 8 bit wide), CADDR (the address bus, 3-bit wide, used for programming the coefficients), LD (the load signal, used for programming the coefficients) and RESET. The only output is OUT (the output bus, 8 bit wide).

During normal operation, the LD and RESET signals are low, the input data streams are provided on IN, one new datum per clock cycle, and the output contains the filtered data, one per clock cycle. The filter coefficients can be reprogrammed by: 1) setting LD; 2) selecting the coefficient with CADDR; 3) providing the coefficient value on IN. One new coefficient can be programmed per clock cycle. During programming, the output does not contain valid data. The coefficients and the internal registers are reset (to zero) by rising the RESET signal. Reset takes one clock cycle.

Although this is a simple design, the programmable filter is complex enough to show the usefulness of our power estimation methodology. First, notice that even during normal operation, successive inputs IN can be strongly correlated, and have widely varying switching activity over time (consider, for example, speech signals, with bursts of sounds and pauses). Moreover, by reprogramming the coefficients, we can completely change the type of filtering performed, and the switching activity. Since reprogramming is expected to be a rather rare occur-

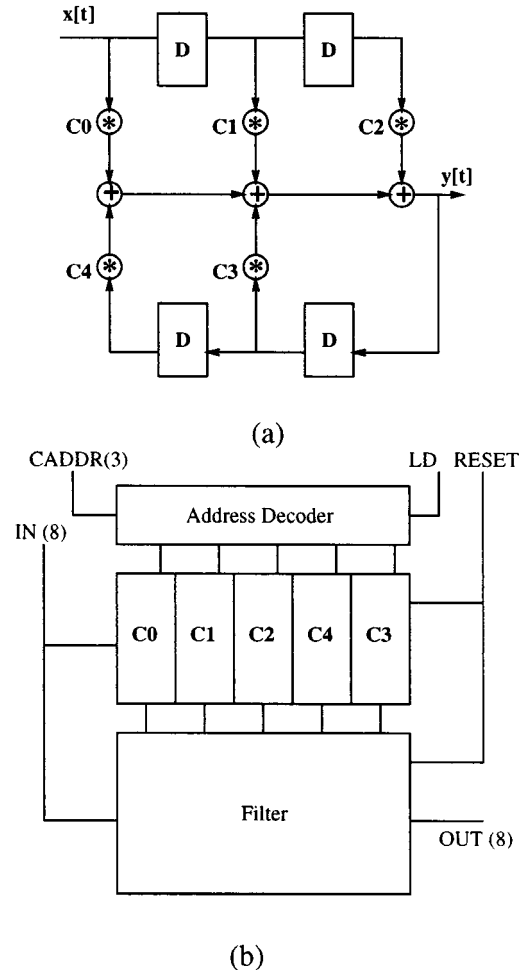


Fig. 7. (a) Flow graph. (b) Architecture of the filter.

rence, it is important to detect when it happens, because power dissipation after reprogramming can change widely.

In our experiments, we created a (long) typical usage stream, including reset and reprogramming phases. Then, we tracked the power dissipation of the filter over time using our power estimation tool. One important characteristics of the design is that it has internal state, and its behavior is determined by

TABLE VII  
 ESTIMATION RESULTS FOR THE FILTER

<i>Actual Power</i>		<i>Estimated Power</i>						
$P_{avg}$	$T_{pow}$	$I(T)$	$P_{avg}^{est}$	$ E $	$RMSRE$	$T$	$SP$	$L2$
2.341	5020	<i>In-Out</i>	2.203	5.894%	0.471	282	17.8	26
		<i>In-Out-State</i>	2.321	0.854%	0.363	323	15.5	31
		<i>Intern</i>	2.333	0.341%	0.347	355	14.1	37

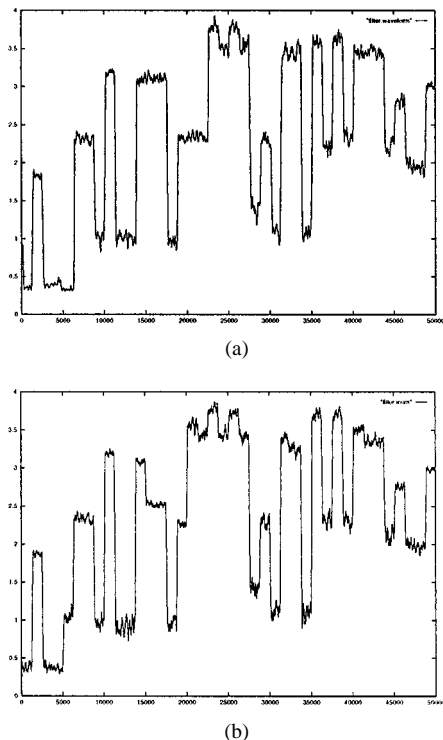


Fig. 8. (a) Mixed smoothing interpolation. (b) Power profile for the filter.

the coefficient values, that change very rarely (and require a maximum of five consecutive clock cycles to be modified).

The filter has been simulated under an input stream consisting of the repeated and interleaved application of a set of patterns to program the coefficients and a burst of input data. Obviously, depending on the type of data to be processed, not all five coefficients need to be reprogrammed. The total length of the stream was 50 000 patterns.

Fig. 8 compares the estimated power waveform to the actual one calculated by exhaustive simulation. For this experiment,  $N_c$  was equal to 147, all other tuning parameters have the same values used for both the combinational and the sequential benchmarks. Table VII collects the results obtained from the application of the input patterns shown on the left of Fig. 8 to the filter.

## VI. CONCLUSION

In this work, we demonstrated how a multilevel simulation engine can be exploited to achieve accurate power estimation in a fraction of the time that would be needed to perform an accurate simulation on the entire pattern stream. Under realistic input stimuli, the average power dissipation of digital systems

is often better described by an up-down staircase function than by a single value. Our multilevel simulation approach achieves high accuracy in tracking how average power varies over time.

A fast simulation of the entire, user provided, input stream is performed. During high-level simulation an indicator function is computed that provides information on when and how often the short-term average power dissipation is expected to change. Low-level accurate simulation is dispatched on small portions of the input stream whenever it is needed for quantitatively tracking how power dissipation is changing over time. When the entire input stream has been processed, only a fraction has been simulated with the slow and accurate power simulator, but the power waveform and the average power are estimated typically within a few percents from the actual ones.

## REFERENCES

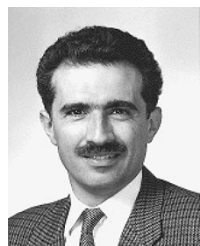
- [1] E. Macii, M. Pedram, and F. Somenzi, "High-level power modeling, estimation, and optimization," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 1061–1079, Nov. 1998.
- [2] F. N. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 446–455, Dec. 1994.
- [3] T. Sato, M. Nagamatsu, and H. Tago, "Power and performance simulator: ESP and its application for 100 MIPS/W class RISC design," in *Proc. ISLPE-94: IEEE Symp. Low-Power Electronics*, San Diego, CA, Oct. 1994, pp. 46–47.
- [4] L. A. Plana and S. M. Nowick, "Architectural optimization for low-power nonpipelined asynchronous systems," *IEEE Trans. VLSI Syst.*, vol. 6, pp. 56–65, Mar. 1998.
- [5] T. Litch and J. Slaton, "StrongARMing portable communications," *IEEE Micro*, vol. 18, no. 2, pp. 48–55, Mar./Apr. 1998.
- [6] S. Rajgopal, "Challenges in low-power microprocessor design," in *Proc. 9th IEEE Int. Conf. VLSI Design*, Bangalore, Karnataka, India, Jan. 1996, pp. 329–330.
- [7] L. Bolcioni, R. Rambaldi, M. Borgatti, R. Guerrieri, and M. Felici, "A low-power, voice-controlled, H.263 video decoder for portable applications," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1810–1819, Nov. 1998.
- [8] S.-Y. Huang, K.-T. Cheng, K.-C. Chen, and M. T.-C. Lee, "A novel methodology for transistor-level power estimation," in *Proc. ISLPED-96: ACM/IEEE Int. Symp. Low-Power Electronics and Design*, Monterey, CA, Aug. 1996, pp. 67–72.
- [9] C.-S. Ding, C.-T. Hsieh, Q. Wu, and M. Pedram, "Stratified random sampling for power estimation," in *Proc. ICCAD-96: IEEE/ACM Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 1996, pp. 577–582.
- [10] R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte-Carlo approach for power estimation," *IEEE Trans. VLSI Systems*, vol. 1, pp. 63–71, Jan. 1993.
- [11] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran," in *Proc. ISCAS-85: IEEE Int. Symp. Circuits and Systems*, Kyoto, Japan, June 1985, pp. 785–794.
- [12] F. Brglez, D. Bryan, and K. Kozmiski, "Combinational profiles of sequential benchmark circuits," in *Proc. ISCAS-89: IEEE Int. Symp. Circuits and Systems*, Portland, OR, May 1989, pp. 1929–1934.
- [13] A. Salz and M. Horowitz, "IRSIM: An incremental MOS switch-level simulator," in *Proc. DAC-26: ACM/IEEE Design Automation Conf.*, Las Vegas, NV, June 1989, pp. 173–178.
- [14] F. Najm, S. Goel, and I. Hajj, "Power estimation in sequential circuits," in *Proc. DAC-32: ACM/IEEE Design Automation Conf.*, San Francisco, CA, June 1995, pp. 635–640.

- [15] V. Saxena, F. Najm, and I. Hajj, "Monte-Carlo approach for power estimation in sequential circuits," in *Proc. EDTC-97: IEEE Eur. Design and Test Conf.*, Paris, France, Mar. 1997, pp. 416–420.
- [16] L. Benini, A. Bogliolo, M. Favalli, and G. De Micheli, "Regression models for behavioral power estimation," presented at the PATMOS-96: Int. Workshop on Power and Timing Modeling, Optimization and Simulation, Bologna, Italy, Sept. 1996.
- [17] M. Nemani and F. Najm, "Towards a high-level power estimation capability," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 588–598, June 1996.
- [18] D. Marculescu, R. Marculescu, and M. Pedram, "Information theoretic measures for power analysis," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 599–609, June 1996.
- [19] C.-T. Hsieh, Q. Wu, C.-S. Ding, and M. Pedram, "Statistical sampling and regression estimation in power macromodeling," in *Proc. ICCAD-96: IEEE/ACM Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 1996, pp. 583–588.
- [20] E. M. Sentovich, K. J. Singh, C. W. Moon, H. Savoj, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Sequential circuits design using synthesis and optimization," in *Proc. ICCD-92: IEEE Int. Conf. Computer Design*, Cambridge, MA, Oct. 1992, pp. 328–333.



**Luca Benini** received the Dr.Eng. degree in electrical engineering from Università di Bologna, Italy, in 1991, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1997, respectively.

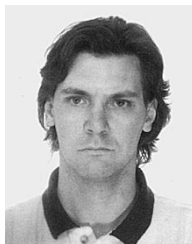
Since 1997, he has been a Research Associate at Università di Bologna and a post-doctoral fellow at Stanford University. He also holds a position as Visiting Scientist at the Hewlett-Packard Laboratories, Palo Alto, CA. His research interests are in all aspects of computer-aided design of digital circuits, with special emphasis on low-power applications.



**Giovanni De Micheli** (S'79–M'79–SM'89–F'94) is professor of electrical engineering, and by courtesy, of computer science at Stanford University, Stanford, CA. His research interests include several aspects of the computer-aided design of integrated circuits and systems, with particular emphasis on automated synthesis, optimization and validation. He is author of *Synthesis and Optimization of Digital Circuits* (New York: McGraw-Hill, 1994); co-author of *Dynamic Power management: Circuit Techniques and CAD Tools* (Norwell, MA: Kluwer, 1998), and

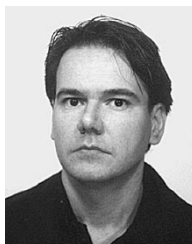
of three other books.

Dr. De Micheli received the 1987 IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS Best Paper Award and two Best Paper Awards at the Design Automation Conference, in 1983 and in 1993. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS.



**Enrico Macii** received the Dr.Eng. degree in electrical engineering from Politecnico di Torino, Turin, Italy, in 1990, the Dr.Sc. degree in computer science from Università di Torino, Turin, Italy, in 1991, and the Ph.D. degree in computer engineering from Politecnico di Torino, in 1995.

From 1991 to 1994, he was an Adjunct Faculty Member at the University of Colorado at Boulder. Currently he is an Associate Professor at the Politecnico di Torino. His research interests include synthesis, verification, and simulation and testing of digital circuits and systems.



**Massimo Poncino** received the Dr.Eng. degree in electrical engineering in 1989 and the Ph.D. degree in computer engineering in 1993, from Politecnico di Torino, Turin, Italy.

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



**Riccardo Scarsi** received the Dr.Eng. degree in electrical engineering from the Politecnico di Torino, Turin, Italy, in 1997. Currently, he is working towards the Ph.D. degree in computer engineering at the same institution.

His research interests include several aspects of the development of algorithms and methods and tools for low-power digital design.