Fast Power Estimation for Deterministic Input Streams *

Luca Benini # Giovanni De Micheli # Enrico Macii ‡ Massimo Poncino ‡ Riccardo Scarsi ‡

Stanford University
Computer Systems Laboratory
Stanford, CA 94305

‡ Politecnico di Torino
Dip. di Automatica e Informatica
Torino, ITALY 10129

Abstract

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 multi-level simulation engine that achieves high accuracy in estimating the average power as well as the time-domain power waveform with high computational efficiency.

1 Introduction

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., streams of validation patterns are usually available or can be produced with a relatively low effort by the designer. Unfortunately, such pattern sets are usually extremely large; hence, performing accurate (and slow) power estimation on the complete sets is simply computationally infeasible.

Recent research [1, 2, 3] has addressed the problem of selecting small subsets of large pattern sets so as to guarantee that the estimate of the average power obtained on the subset is as close as possible to the actual average power dissipation of the circuit when simulated with the complete pattern set.

In this paper, we tackle the problem of input pattern selection from a novel point of view. Based on the observation of real-life systems, we claim that the average power does not fully characterize the power dissipation of a circuit, and that a *time-domain power waveform* can provide a much more complete source of information. We move from the experimental evidence that typical, deterministic input streams are such that the average power dissipated by the circuit can vary widely when distinct subsets of consecutive patterns are simulated. We thus propose a technique, leveraging multi-level simulation, that can be used to quickly, yet accurately extract the average power, as well as the power waveform for deterministic input streams for which the *running average*, taken over a fixed number input patterns, of the circuit power dissipation follows an "up-down staircase" curve.

From the user point of view, a fast cycle-based simulation is performed on the complete input stream. During the simulation, the monitoring of an *indicator function* 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. The tool automatically constructs the time-domain power waveform and optionally computes its average.

*This work is partially supported by NSF under contract MIP-9421129

The indicator function may not 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 trade-off between the accuracy of such function and the overhead it imposes on cycle-based simulation. Our method is practical only if it guarantees that the time spent on fast cycle-based simulation (T_c), plus the overhead for the computation of the indicator function (T_{ind}), plus the time required to perform accurate power simulation on the the sample (T_{sample}) is much smaller than the time required to perform accurate power simulation on the entire input stream (T_{pow}).

We have benchmarked the capabilities of our tool on large subsets of the ISCAS'85 [4] and the ISCAS'89 [5] circuit suites. In particular, 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 [6] of the entire input stream. Results are highly satisfactory, since the average error is between 4.5% and 6.1% for the combinational examples, and between 2.2% and 6.4% for the sequential circuits, depending on the type of indicator function used. Moreover, the estimated power waveforms match closely the actual ones.

2 Related Work

There are two techniques in the literature which are somehow related to ours. The power ratio method [7] is based on multi-level simulation. It postulates that the power estimate, P_{est}, provided by high-level, fast power simulation is proportional to the power, P_{acc}, estimated with accurate simulation. In symbols: $P_{acc} = P_{est} \cdot K$. The unknown proportionality constant, K, is simply obtained by first computing both P_{acc} and P_{est} for a small subset of the complete input stream, and then by taking their ratio. Unfortunately, for deterministic, designer-supplied streams, K is usually far from being constant; in these cases, the method may lead to inaccurate estimates.

Stratified random sampling [8], though developed as an alternative to Monte Carlo methods, has some similarities to the technique we propose here. 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, it 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 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.

3 Tracking Staircase 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 non-monotonic fashion, with long plateaus where $P_N(T)$ is approximately constant mixed with regions where it changes rapidly. Figure 1 (a) shows the diagram of a time-varying property $P(T)$ with staircase behavior, and Figure 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.

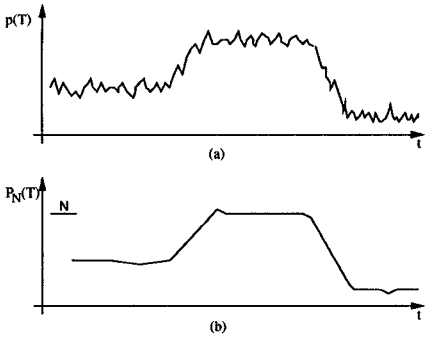

Figure 1: Waveforms for $P(T)$ (a) and $P_N(T)$ (b).

The power dissipated by many real-life circuits, such as microprocessors running realistic work-loads, has staircase behavior [9]. 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. Extensive experimentation [10, 11] has shown that, when the input streams are randomly generated, composed by uniformly distributed, uncorrelated patterns, the power dissipation does not have a staircase behavior. On the contrary, $P_N(T)$ is a roughly constant function, as long as N is sufficiently large to smooth out cycle-by-cycle variations due to the strong pattern dependency of power. Obviously, we would like to accurately estimate the power dissipation induced in a circuit by realistic input streams which are not uniformly distributed, nor uncorrelated. Most of the pattern sets usually provided by the designers for typical hardware modules (e.g., RT-level macros, cores, microcontrollers) tend to induce a staircase behavior on the power dissipation.

Our target is a challenging one, because it requires a difficult trade-off 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 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.

3.1 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 much milder requirement than simple proportionality (as requested by the power ratio method).

We propose a set of alternative indicator functions based on sampling the (zero-delay) switching activity and the value of the nodes in Level 1 simulation. Tuning the number and choice of the sampling points allows us to:

- Trade off 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 [12], it was observed that there exists strong correlation between power dissipation in a combinational logic circuit and the switching activity at its inputs and outputs. A similar conclusion was reached in [13, 14]; in these works, the power dissipated by a combinational design, possibly described at the functional-level, was approximated by a combination of the circuit input and output *entropy*, two quantities which are strictly related to the input and output switching activities.

The basic claim in [12] is that power dissipation can be predicted with reasonable accuracy by computing a function of input-output switching. In symbols:

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

where P^{n+1} is the power dissipated during the $n+1$-st simulation cycle, i_k^n and i_k^{n+1} are respectively the value 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 non-linear) mapping $f : B^{n_i+n_o} \rightarrow \mathcal{R}$. The choice of the sampling points for our indicator functions is based on a similar assumption. We propose the following criteria, sorted by 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 considerably 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 very 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 different level of abstraction, because the values of the internal nodes cannot be sampled (they do not exist at higher abstraction). 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 sampling criteria listed above span the trade-off between completeness of the monitored information and computational overhead. It should be noted that the overhead strongly depends on the available Level 1 simulation. State-of-the-art fast cycle-based simulators have the capability of collapsing internal nodes or using compact data representations to accelerate the computation. The overhead can be estimated based on how much of this acceleration capability must be given up.

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. For this reason, our indicator function must include a short-term time averaging operation. The simplest one is then the number of switching events averaged over a short number of cycles, 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, ρ_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 Equation 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 4.

Notice that 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 can be replaced by more complex indicator functions. The one we have adopted computes the weighted sum of switching events averaged over 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, ρ_i^t of Equation 3 is replaced by $w_i \rho_i^t$. The weights w_i represent the relative influence of the switching of sampling point i. Several algorithms can be devised for the computation of w_i (notice that such algorithms must be run only once at the beginning of the estimation process). Possible choices of w_i are:

1. Expected capacitance (based on node fanout) for internal nodes when the most expensive choice of sampling points is used;
2. For primary outputs and flip flop inputs: Estimate of the complexity (i.e., number of gate equivalent) in the fanin cone;
3. For primary inputs and flip-flop outputs: Estimate of the complexity (i.e., number of gate equivalent) in the fanout cone.

A generic, user-specified non-linear (weighted) combination of values and transitions averaged over N_c might also be adopted. However, this indicator function is not investigated here, since we target a fully automated power estimation process.

Concluding the section, we want to stress that the fundamental use of any indicator function $I(T)$ is *not* to provide an estimate of the power consumption at Level 1 (such estimates have limited accuracy and may be interesting only during design exploration). Instead, we use $I(T)$ to extract information on the variations over time of the average power dissipation. How this can be done and how such information is fruitfully exploited to determine accurate power estimates is the subject of the next section.

4 Multi-Level Power Simulation

Our multi-level simulation engine is conceptually simple. Given a long stream of N_{tot} input patterns, a high-level and a low-level description of the circuit under analysis, the Level 1 simulation is started. Level 1 simulation runs three additional tasks:

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 choice of the sampling points and the computation of $I(T)$ have been discussed in Section 3. The purpose of the staircase test is to decide, based on the changes over time of $I(T)$, when to fire the Level 2 simulation. Whenever the staircase test triggers, the inputs and state values are extracted from Level 1 information and used to set the initial state for Level 2 simulation. After that, the multi-level 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 $P_{avg}(T)$ (notice that P_{avg} is a function of time). The value $P_{avg}(T)$ 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_{tot} patterns have been simulated at Level 1, but only N_{sample} patterns have been simulated at Level 2. Typically, $N_{sample} << N_{tot}$. The end results is a set of values $\mathcal{P} = \{P_{avg}(T_1), ..., P_{avg}(T_s)\}$, where $T_1 < ... < T_s$. The power values $P_{avg}(T_i)$ and the times T_i are a set of samples of the *power waveform* of the circuit.

The performance of our multi-level 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 on systems with staircase behavior.

The speed-up is defined by the following equation:

$$SP = \frac{T_{pow}}{T_c + T_{ind} + T_{sample}} \quad (4)$$

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

The accuracy of the multi-level scheme can be measured by comparing the average power, P_{avg}, obtained by simulating the complete input stream at Level 2 with the estimate, P_{avg}^{est}, obtained by averaging the power samples $P_{avg}(T_i)$. Clearly, the latter must be weighted with the *duration* of the samples: $P_{avg}^{est} = \sum_{i=1}^{s} \tau_i P_{avg}(T_i)$. The duration τ_i of $P_{avg}(T_i)$ is defined as:

$$\tau_i = \frac{T_i - T_{i+1}}{N_{tot} T_{clk}} \quad (5)$$

where T_{clk} is the clock period of the system. 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.

The definition of our multi-level estimation strategy is completed by addressing the following three open issues:

- How to choose N_c, i.e., the length of the short-term, 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).

4.1 Choosing N_c

The choice of the parameter N_c, that is, the length of the sliding window for the running average, is a key point for the success of the entire strategy. 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 multi-level 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 multi-level simulation becomes really 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 multi-level 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 $N_c = kN_{M_I}$, $0 < k \leq 1$. 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 (or even larger, if we suspect extremely long plateaus and we have huge N_{tot}). 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.

4.2 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 $I(T)$; the procedure for its application is the following:

- Level 2 simulation is first started to obtain the first power sample. Concurrently, $I(T)$ 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 to $T > T_0$. The staircase test is not applied until $T = T + \beta T_{clock} N_c$, 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. The interval during which the test is disabled is called *recovery interval*.
- After the recovery interval, the test is applied at every clock cycle T. A staircase behavior is detected when:

$$\Delta|I(T)| > \gamma \Delta_{max}|I| \quad (6)$$

  where $\Delta|I(T)| = |I(T_0) - I(T)|$, γ 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 Equation 2, $\Delta_{max} = N_c|S|$, 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, the power is estimated through Level 2 simulation. Upon stopping of the latter (at time T_2), $I(T_2)$ is stored and the process is restarted.

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

4.3 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 non-staircase, 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: The 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 on the indicator function, although it is expected to be slower.

Concluding this section, we would like to emphasize that, although no strong convergence results or accuracy bounds are available for our technique, this limitation is shared by all other methods presented in the literature when applied to realistic power estimation problems (i.e., estimation of the power dissipation for sequential systems with user-provided input streams). Experimental evidence, detailed in Section 5, confirms the robustness and flexibility of our approach, which appears to be a viable, albeit heuristic, power estimation strategy for the realistic staircase situation. More work is needed to gain a complete theoretical understanding of the mathematical models for power dissipation in this general setting.

5 Experimental Results

In order to investigate the applicability and the accuracy of our 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 [6]. In the following sections we report on our findings.

The algorithm has shown remarkable robustness for what concerns the values of the user-defined tuning parameters that control the performance and accuracy. To select such values we have ran a set of experiments on a small combinational circuit (a few tens of gates) and we have chosen the values that have given the minimum error (on average power estimation) for an overall 100X speed-up. The parameter values are the following: $k = 0.5$, $\beta = 1$, $\gamma = 0.2$, $\eta_L = 0.5$ and $\eta_U = 2$. The same values have been used for all experiments described in the following three sections.

In spite of the satisfactory results we have obtained, we conjecture that even more accurate (or faster) estimates could be obtained by tuning the parameters for each example. However, this would greatly increase the time required to obtain the power estimates (parameter tuning should be taken into account when measuring the speed-up).

5.1 Combinational Circuits

In a first set of experiments, we have considered all the ISCAS'85 examples [4]. The circuits were first optimized for area, and then mapped for area using SIS [15] onto a gate-library consisting of 2-input NAND and NOR gates, inverters, and buffers.

Table 1 reports the data for the ten considered benchmarks. Columns *Circ*, *I*, *O*, and *G* give the number of circuit inputs, outputs, and gates. Column *Estimated Power* shows the average power values obtained with the technique of this paper, while column *Actual Power* provides the values obtained through complete transistor-level simulation of the input stream. Column *Estimated Power* is further split in three sets of columns, one for each of the three sampling points selection criteria we have considered, namely, input (column *In*), input/output (column *In-Out*), and global activity (column *Intern*). Each column reports the average power (*Pow*), the absolute value of the relative error, with respect to the actual power ($|E|$), the number of Level 2 simulations triggered (*L2*), and the time required to complete the estimation (T), measured in seconds on a DEC AXP 1000/400 with 256 MB of RAM. Finally, column N_c tells the size of the sliding window N_c, specific for each circuit, that we have used in the indicator function. All power values in this, and in the following tables, are expressed in mW.

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 correlation (i.e., correlation between successive patterns), very high spatial correlation (i.e., correlation between input variables) and, more importantly, are highly non-stationary. The average input switching activity is changed abruptly and wide variations are imposed several times in the stream. Between variations, the average input switching activity is roughly constant, although the vectors are correlated. The complete streams consisted of 50,000 patterns.

The error between the power values provided by our tool and the ones obtained through complete transistor-level simulation of the input stream are very limited, namely, 6.1%, 5.9%, and 4.5%, depending on the selected sampling criterion. On the other hand, the simulation time reduction is substantial (between one and two orders of magnitude).

As expected, the choice of a more accurate criterion for sample points selection yields increasingly accurate estimates. However, it should be observed that the difference in accuracy is quite small. This is of great interest when the 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.

Figure 2 shows, in the case of 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 input stream. The piecewise-constant approximation of the power waveform, obtained with the indicator function based on I/O sampling, 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 two, since the indicator function provide numbers of weighted switching events, as opposed to actual power values. Comparing the two bottom plots, it can be observed that the indicator function based on I/O activity provides sufficient information to track the staircase behavior of the power for combinational circuits. Notice, however, that we approximate the power with a piecewise-constant waveform, hence the cycle-by-cycle accuracy is limited. More refined interpolation schemes can be used to improve cycle-by-cycle accuracy [16].

Circ	I	O	G	Estimated Power												N_c	Actual Power	
				In				In-Out				Intern					Pow	T
				Pow	$\|E\|$	L2	T	Pow	$\|E\|$	L2	T	Pow	$\|E\|$	L2	T			
c432	37	6	265	0.72	7.5	15	9	0.72	7.5	15	9	0.71	6.4	15	9	158	0.67	2125
c499	41	32	525	1.33	6.4	13	10	1.36	8.8	11	8	1.30	4.0	15	11	110	1.25	2281
c880	60	26	431	1.42	9.2	16	11	1.42	9.2	15	11	1.41	8.4	13	9	91	1.30	2160
c1355	41	32	525	1.38	2.2	9	6	1.38	2.2	9	6	1.38	2.2	8	5	220	1.35	4813
c1908	33	25	529	1.78	2.3	13	21	1.75	1.1	16	30	1.74	0.9	13	24	197	1.73	7804
c2670	233	140	808	2.69	0.4	13	21	2.69	0.4	13	21	2.69	0.4	13	21	112	2.68	9483
c3540	50	22	1289	5.43	4.4	12	252	5.31	2.1	15	301	5.14	1.1	12	246	311	5.20	12102
c5315	178	123	1759	6.50	1.7	12	170	6.52	2.0	13	171	6.50	1.7	19	311	303	6.39	10209
c6288	32	32	3240	9.79	8.1	9	3237	9.81	7.9	9	3316	9.96	6.5	8	3675	307	10.66	32663
c7552	207	108	2314	10.14	18.9	8	1432	10.19	18.5	16	1560	10.82	13.6	15	1470	425	12.51	10812
Average					6.1				5.9				4.5					

Table 1: Power Estimation Results for Combinational Circuits.

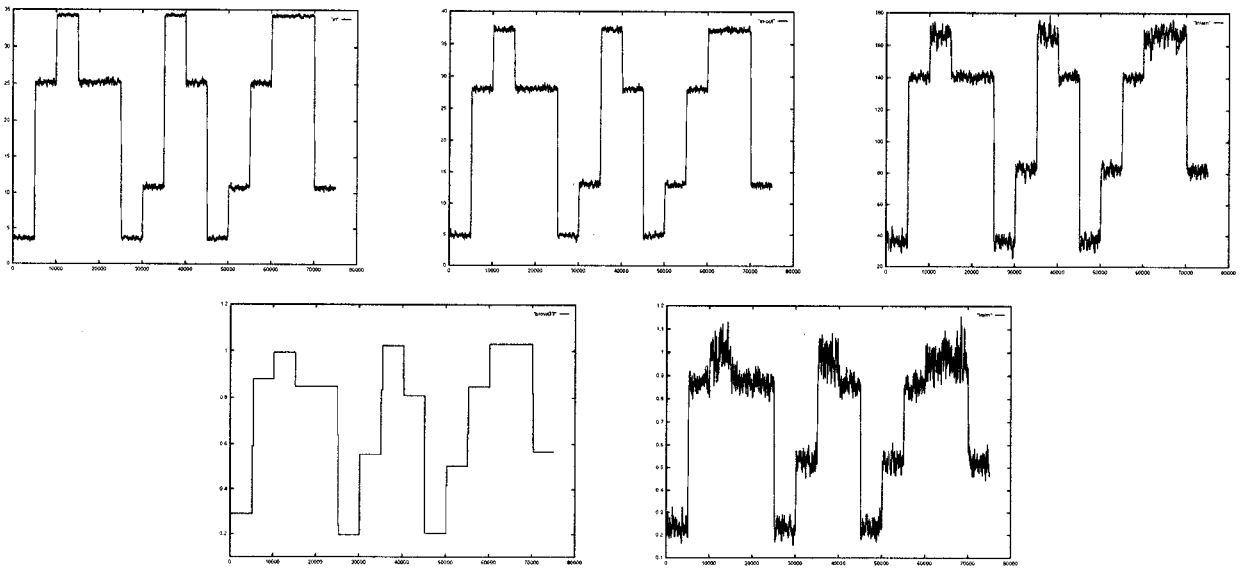

Figure 2: Indicator Functions, Piecewise-Constant Approximation, and Actual Power Profile for Benchmark c432.

5.2 Sequential Circuits

The second set of experiments concerns sequential circuits. We have selected a subset of the ISCAS'89 benchmarks [5], 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.

Table 2 shows the results obtained for these circuits (column *FF* indicates the number of flip-flops). Notice that for these experiments the indicator functions we have used are the following: i) primary input and primary output activity (column *In-Out*); ii) primary input, primary output, and state (i.e. flip-flop inputs and outputs) activity (column *In-Out-State*); iii) complete internal activity (column *Intern*).

As for the combinational examples, our method yields accurate estimates, the average errors being 6.4%, 3.5%, and 2.2%, depending on the sampling points selection approach. Execution times are still much smaller (about one order of magnitude) than those required by exhaustive simulation.

For some sequential circuit topologies, the temporal behavior of the indicator function based solely on input-output switching is not necessarily informative enough on the actual power. This is because, for sequential circuits having far more latches than primary inputs, the degree of switching at the primary inputs has a limited correlation to the overall activity (and thus power); rather, this is mainly driven by the switching at the present state lines, which induces most of the switching inside the combinational logic.

Figure 3 shows a graphical view of this fact. The plots at the top of the figure refer to circuit **s298**. This circuit clearly 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, our tool tracks the behavior of the power dissipation very well, because the indicator function observes the switching activity of the state variables as well. The piecewise-constant approximation 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). Power has a staircase behavior that is easily tracked by our tool.

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 even by our 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).

Circ	I	O	FF	G	Estimated Power												N_c	Actual Power	
					In-Out				In-Out-State				Intern					Pow	T
					Pow	$\|E\|$	L2	T	Pow	$\|E\|$	L2	T	Pow	$\|E\|$	L2	T			
s298	3	6	14	136	0.045	9.6	20	30	0.048	2.8	24	31	0.048	2.6	31	34	115	0.050	293
s344	9	11	15	151	0.080	2.2	10	21	0.080	2.2	10	21	0.081	1.9	14	28	122	0.082	454
s420	18	1	16	168	0.035	12.7	12	20	0.038	6.1	14	20	0.039	4.4	15	23	125	0.040	272
s444	3	6	21	181	0.038	16.5	10	14	0.042	5.1	12	29	0.046	0.8	23	26	123	0.046	136
s510	19	7	6	268	0.095	8.2	15	43	0.091	4.6	17	41	0.098	3.5	27	33	131	0.095	225
s713	35	23	17	189	0.061	3.9	13	25	0.061	4.4	14	27	0.060	2.2	16	29	142	0.058	422
s1196	14	14	18	647	0.189	5.0	12	67	0.188	4.4	12	66	0.185	2.7	13	80	115	0.180	956
s1423	17	5	74	757	0.250	1.5	12	83	0.250	1.4	24	172	0.253	0.3	13	179	180	0.254	825
s5378	35	49	164	1585	0.444	0.7	14	198	0.444	0.7	13	154	0.444	0.7	15	208	154	0.447	1513
s13207	31	121	669	2693	0.515	4.3	12	85	0.510	3.4	12	89	0.510	3.2	12	90	163	0.493	1770
Average						6.4				3.5				2.2					

Table 2: Power Estimation Results for Sequential Circuits.

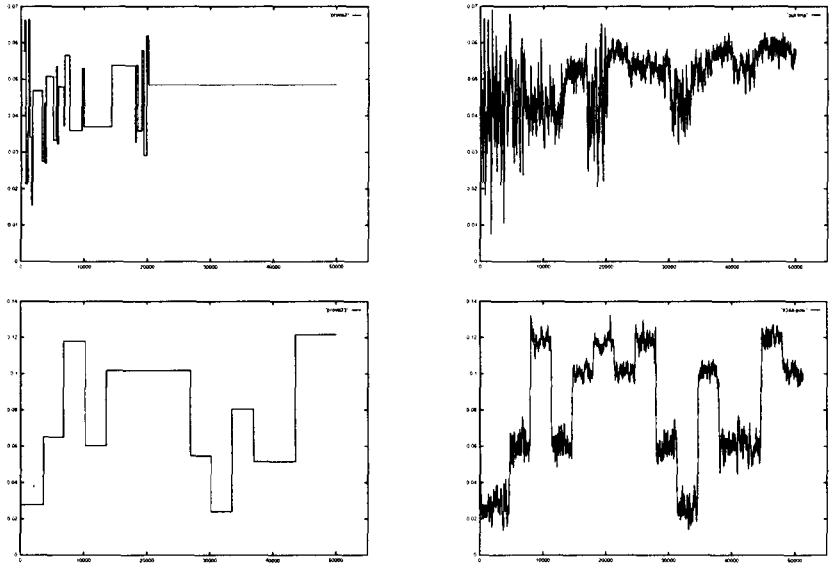

Figure 3: Different Behavior of Two Sequential Benchmarks.

It is worth observing that the accuracy of the average power estimates for sequential circuits is not deteriorated with respect to the combinational ones. Average errors are indeed inferior, although this is probably due to the nature of the benchmarks. In the general case, we expect similar accuracy when an Input-Output-based $I(T)$ is used for combinational circuits and an Input-Output-State-based $I(T)$ is used for sequential circuits.

5.3 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 in a $2\mu m$ CMOS technology. The design contained 2190 gates (approximately 4000 transistors).

The flow graph of the filter is shown in Figure 4 (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 Figure 4 (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).

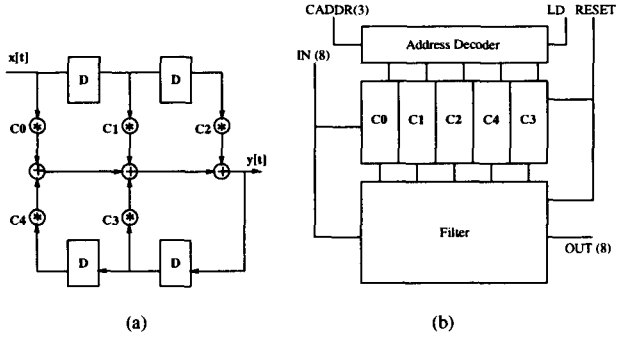

Figure 4: Flow Graph (a) and Architecture (b) of the Filter.

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 re-programmed by: i) Setting LD; ii) Selecting the coefficient with CADDR; iii) 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 0) by rising the RESET signal. Reset takes one clock cycle.

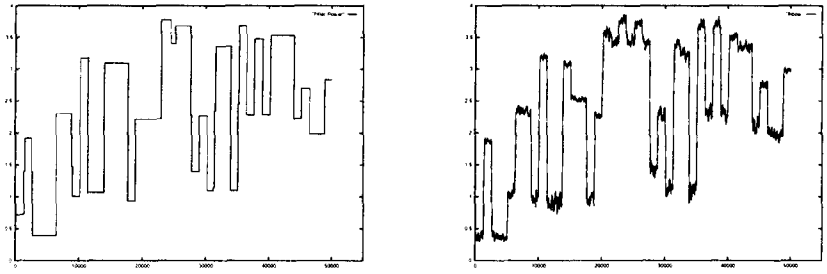

Figure 5: Piecewise-Constant Approximation (Left) and Actual Power Profile (Right) for the Simulated Input Stream.

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 re-programming the coefficients, we can completely change the type of filtering performed, and the switching activity. Since re-programming is expected to be a rather rare occurrence, it is important to detect when it happens, because power dissipation after re-programming can change widely.

In our experiments, we created a (long) typical usage stream, including reset and re-programming 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 is has internal state, and its behavior is determined by the coefficient values, that change very rarely (and require a maximum of 5 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 re-programmed. The total length of the stream was 50,000 patterns. Figure 5 compares the estimated power waveform of the input stream to the actual power waveform 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 3 shows the power values obtained from the application of the input patterns of Figure 5 to the filter.

Estimated Power					Actual Power	
I(T)	Pow	E	L2	T	Pow	T
In-Out	2.362	4.4	24	513	2.262	5020
In-Out-State	2.351	3.9	23	442		
Intern	2.334	3.1	20	382		

Table 3: Power Estimation Results for the Programmable Filter.

6 Conclusions and Future Work

In this work we demonstrated how a multi-level simulation engine can be exploited to achieve accurate power estimation in a small 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 multi-level 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 simulated, only a small 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.

Although we provided extensive experimental evidence of the efficiency and accuracy of our approach (on both standard benchmarks and a realistic case study) more work needs to be done in two directions. First, we plan to apply our tool to much larger systems such as microprocessors and DSP chips, where typical input streams are actual user applications and software benchmarks. Second, we want to investigate the theoretical properties and the domain of applicability of our method.

References

[1] D. Marculescu, R. Marculescu, M. Pedram, "Stochastic Sequential Machine Synthesis Targeting Constrained Sequence Generation," DAC-33, pp. 696-701, Las Vegas, NV, June 1996.

[2] R. Marculescu, D. Marculescu, M. Pedram, "Adaptive Models for Input Data Compaction for Power Simulators," ASPDAC-2, pp. 391-396, Chiba, Japan, January 1997.

[3] R. Marculescu, D. Marculescu, M. Pedram, "Hierarchical Sequence Compaction for Power Estimation," DAC-34, pp. 570-575, Anaheim, CA, June 1997.

[4] F. Brglez, H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," ISCAS-85, Kyoto, Japan, June 1985.

[5] F. Brglez, D. Bryan, K. Kożmiński, "Combinational Profiles of Sequential Benchmark Circuits," ISCAS-89, pp. 1929-1934, Portland, OR, May 1989.

[6] A. Salz, M. Horowitz, "IRSIM: An Incremental MOS Switch-Level Simulator," DAC-26, pp. 173-178, Las Vegas, NV, June 1989.

[7] S-Y. Huang, K-T. Cheng, K-C. Chen M. T-C. Lee, "A Novel Methodology for Transistor-Level Power Estimation," ISLPED-96, pp. 67-72, Monterey, CA, August 1996.

[8] C-S. Ding, C-T. Hsieh, Q. Wu, M. Pedram, "Stratified Random Sampling for Power Estimation," ICCAD-96 San Jose, CA, pp. 577-582, November 1996.

[9] D. Singh, "Prospects for Low Power Microprocessor Design," IWLPD-94, Keynote Address, pp. 1, Napa Valley, CA, April 1994.

[10] F. Najm, S. Goel, I. Hajj, "Power Estimation in Sequential Circuits," DAC-32, pp. 635-640, San Francisco, CA, June 1995.

[11] V. Saxena, F. Najm, I. Hajj, "Monte-Carlo Approach for Power Estimation in Sequential Circuits," EDTC-97, pp. 416-420, Paris, France, March 1997.

[12] L. Benini, A. Bogliolo, M. Favalli, G. De Micheli, "Regression Models for Behavioral Power Estimation," PATMOS-96, pp. 179-187, Bologna, Italy, September 1996.

[13] M. Nemani, F. Najm, "Towards a High-Level Power Estimation Capability," IEEE Trans. on CAD, Vol. 15, No. 6, pp. 588-598, June 1996.

[14] D. Marculescu, R. Marculescu, M. Pedram, "Information Theoretic Measures For Power Analysis," IEEE Trans. on CAD, Vol. 15, No. 6, pp. 599-609, June 1996.

[15] E. M. Sentovich, K. J. Singh, C. W. Moon, H. Savoj, R. K. Brayton, A. Sangiovanni-Vincentelli, "Sequential Circuits Design Using Synthesis and Optimization," ICCD-92, pp. 328-333, Cambridge, MA, October 1992.

[16] L. Benini, G. De Micheli, E. Macii, M. Poncino, R. Scarsi, "Quick Generation of Temporal Power Waveforms for RT-Level Hard Macros," ISIS-97, Austin, TX, October 1997, To Appear.