

# Power Estimation of Cell-Based CMOS Circuits

Alessandro Bogliolo

Luca Benini

Bruno Riccò†

CSL - Stanford University  
Stanford, CA-94305

†DEIS - University of Bologna  
Bologna, I-40136

## Abstract

*PPP is a Web-based simulation and synthesis environment for low-power design. In this paper we describe the gate-level simulation engine of PPP, that achieves accuracy always within 6% from SPICE, while keeping performance competitive with traditional gate-level simulation. This is done by using advanced symbolic models of the basic library cells, that exploit the physical understanding of the main power-consuming phenomena. VERILOG-XL is used as simulation platform to maintain compatibility with design environments. The Web-based interface allows the user to remotely access and execute the simulator using his/her own Web-browser (without the need of any software installation).*

## 1 Introduction

Power dissipation in VLSI systems has recently become a critical metric for design evaluation. A large number of power estimation techniques has been recently proposed [3, 4, 5, 6], based on models at different levels of abstraction, ranging from electrical level to architectural level [7, 8, 9]. Electrical-level simulators produce the most accurate results, but are often very demanding in terms of computational resources. Moreover, the large number of simulations needed to reach a reliable estimate of average power dissipation further restricts the class of circuits that can be analyzed with electrical simulators in a reasonable time.

At a higher level of abstraction, logic-level simulation allows power estimation for very large blocks, often enabling full-chip simulation. As a consequence, for CMOS digital circuits, logic-level simulation is usually the preferred tool for validation and debugging. In the simplest logic-level model, power is estimated observing the *switching activity* at the output of the basic logic blocks of the circuit (weighted by the load capacitance). The advantage of this model is that it enables the application of *pattern independent* techniques, which provide an estimate of the average switching activity without actually simulating the circuit [4].

Power estimation based on switching activity has however limited accuracy, mainly because it does not consider phenomena such as non-instantaneous signal transitions, spu-

rious transitions (glitches) and gate internal capacitances, that may have a sizable impact on the total power dissipation. In order to overcome these difficulties, advanced logic simulation techniques have been proposed that allow increased accuracy, while maintaining the abstraction at the logic level [11, 12, 13]. In these approaches, lookup tables are obtained by electrical simulation of the basic building blocks, and the collected data is then used during gate-level simulation.

Although these techniques reported promising results, they have two main limitations. First, they do not assume any model for the internal structure of the basic building blocks (*gates*). Second, they do not deal with multiple input transitions that are not perfectly aligned in time. In this work we propose a more accurate model (based on the physical understanding of the electrical phenomena involved in power dissipation) that overcomes the limitations above mentioned, while keeping computational efficiency competitive with traditional gate-level power simulation.

Our technique exploits a BDD-based symbolic model for describing the charge and discharge of parasitic (and load) capacitances and the flow of short circuit current. Lookup tables are used only for modeling the timing behavior of the circuit, as it is commonly done in full-delay simulation. Our model is flexible and can be used to accurately estimate power dissipation for gates in a large range of load and input conditions. As a result, our method is highly accurate also for single gate (*local*) power estimate, allowing the individuation of critical gates during design optimization.

We have implemented our method using simulator VERILOG-XL. For our test library the accuracy on *local* power estimation is within 6% from SPICE under a wide range of fan-in and fan-out conditions, while the accuracy on the average power dissipation for large benchmarks is even higher. The speed penalty with respect to unit-delay VERILOG simulation is within a factor of 6, while the speedup with respect to fast SPICE simulation ranges from two to three orders of magnitude.

We have embedded our simulator into PPP [1, 2], a new Web-based simulation environment for CMOS integrated circuits. Thanks to its highly interactive Web-based interface, PPP can be remotely accessed and executed through the network using traditional web-browsers.

## 2 Gate-Level Power Simulation

Traditional gate-level power estimates are based on the simplifying assumption that the supply current required by a

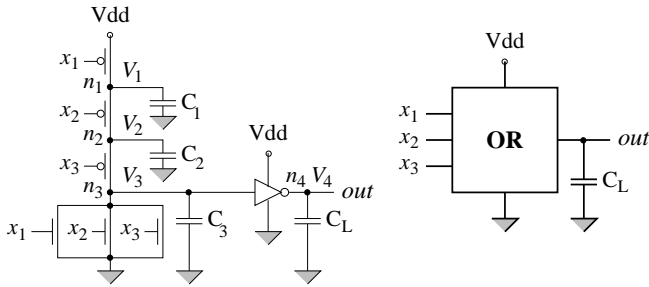


Figure 1: CMOS realization of a three input OR gate. Parasitics are modeled by means of four constant capacitors to ground:  $C_1 = C_2 = 11\text{ fF}$ ,  $C_3 = 157\text{ fF}$  and  $C_L = 136\text{ fF}$ .

CMOS circuit is essentially spent on charging load capacitances at the outputs of the switching gates. Because of this assumption, the inner structure of the gates is neglected and the average power consumption is evaluated simply by looking at the switching activity (toggle count) and the capacitive load at the gate outputs.

In this way, however, the actual power consumption can be heavily underestimated since several parasitic phenomena (such as short-circuit currents, charging and discharging of internal capacitances and charge sharing) that may have a sizable impact on the global power, cannot be captured.

Moreover, spurious transitions (glitches) that may represent the 20% of the switching activity [10], cannot be accurately accounted for, due to the use of simplified cell delay models.

**Example 1** Fig. 1 represents a CMOS realization of a three inputs OR gate. At a logic level of abstraction, changing the inputs from  $\mathbf{x} = 100$  to  $\mathbf{x} = 010$  does not cause any effect (boldfacing is hereafter used to denote Boolean vectors:  $\mathbf{x} = [x_1 x_2 x_3]$ ). However, a misalignment between the falling and rising edges of input signals  $x_1$  and  $x_2$  (i.e.,  $x_2$  rising 0.4ns after  $x_1$  has fallen), may give rise to a sizable power consumption because of two phenomena: *i*) a double, spurious transition (glitch) at the output node, causing the charging/discharging of both  $C_L$  and  $C_3$ , and *ii*) short-circuit currents through both the CMOS stages. The power consumption provided by Spice for a 20ns transient analysis is of 0.08mW.  $\square$

In principle, however, the above mentioned limitations can be overcome by taking advantage, during gate-level simulation, of previously collected information about the basic building blocks of the circuit.

## 2.1 Cell-Based Power Simulation

Cell-based power estimation [11] consists of two steps: cell characterization and logic simulation. The characterization phase includes a set of electrical simulations of each library-cell for all possible input transitions and for a wide range of fan-in and fan-out conditions. Timing and power information obtained in this way is used to construct lookup tables. Logic simulation is then performed by a back-annotated event-driven simulator. Whenever a transition occurs at the input of a gate, both the propagation delay and the energy

drawn by the gate are read on the corresponding lookup tables.

In principle, as long as the lookup tables have entries corresponding to the actual fan-in/fan-out conditions of each gate of the circuit, the back-annotated gate-level simulation reaches the accuracy of the electrical one. In practice, however, accuracy must be traded-off with the lookup table size. Hence, power and delay are precharacterized only for single input transitions and for a small set of typical values of the I/O parameters (input slopes and output loads). The subsequent discretization ultimately impairs the accuracy of the power estimate. Moreover, input skews and internal charge status are usually neglected, giving rise to further approximations.

**Example 2** With respect to Fig. 1, consider a transition from  $\mathbf{x} = 101$  to  $\mathbf{x} = 001$ . The actual energy drawn by the OR gate corresponding to this input transition depends on the charge status of its internal capacitances. In particular, no supply energy is required if both  $C_1$  and  $C_2$  have already been charged at  $V_{dd}$  by a previously applied input vector  $\mathbf{x} = 001$ , while otherwise 0.44pJ ( $22\mu\text{W}$  with a 20ns cycle-time) are dissipated.  $\square$

Recently, more refined methods have been proposed that partially exploit the knowledge of the power consuming phenomena inside the cells. In [13] it is observed that the power dissipated by a cell is characterized by two radically different behaviors depending on the ratio between input and output slopes. Two fitting formulas are then used. The accuracy in this approach is limited by the simple analytic model and by the lack of information on the internal charge state.

In [12] a finite-state machine model for the cell is proposed, in which the internal charge status of the gate is modeled, and the power dissipated during input transitions is represented by weights associated with the state transitions of the FSM. However, the dependence of power consumption on capacitive load and signal slopes is not explicitly modeled, thus requiring the use of (large) lookup tables associated with each transition.

Moreover, both techniques do not model accurately the effect of multiple misaligned input transitions and parasitic phenomena causing intermediate voltage levels (such as signal glitches and charge sharing). In the next section, we present a new cell-based approach that overcomes the above limitations.

## 3 A New Approach

As mentioned before, cell-based power estimation techniques must compromise between the size of the lookup tables and the accuracy of the estimates. In order to obtain a better trade-off between efficiency and accuracy, we use an advanced model of power consumption in CMOS cells, based on the understanding of the electrical phenomena involved.

### 3.1 Fundamental Idea

In principle, the supply current  $I(t)$  drawn by a CMOS cell corresponding to an input transition (namely, from  $\mathbf{x}^i$  to  $\mathbf{x}^f$ ) can be viewed as consisting of two contributions:

- a **charging** current  $I_c(t)$ , that increases the total charge of internal and load capacitors, and
- a **wasted** current  $I_w(t)$ , that does not affect the charge status of the cell.

The energy drawn by the cell during the whole transition can accordingly be partitioned into two contributions:

$$E_c = \int_{t^i}^{t^f} V_{dd} I_c(t) dt, \quad E_w = \int_{t^i}^{t^f} V_{dd} I_w(t) dt.$$

Apexes  $i$  and  $f$  are hereafter used to denote the beginning and the end of a given transition, respectively.

In general, even if we run accurate electrical simulations, we cannot distinguish between the two portions of supply current ( $I_c$  and  $I_w$ ). However, we can evaluate  $E_c$  simply by looking at the charge status of the cell. In fact,

$$E_c = V_{dd} \int_{t^i}^{t^f} I_c(t) dt = V_{dd} \Delta Q, \quad (1)$$

where  $\Delta Q$  is the total charge provided by  $V_{dd}$  to internal and load capacitors.  $\Delta Q$  can therefore be computed by the following equation:

$$\Delta Q = \sum_{j \in \mathcal{S}} \Delta q_j, \quad (2)$$

where  $\mathcal{S}$  is the set of nodes with a connection to  $V_{dd}$  for input vector  $\mathbf{x}^f$ , and  $\Delta q_j$  is the charge increase at node  $n_j$ :  $\Delta q_j = C_j (V_j^f - V_j^i)$ . It is worth noting that  $E_c$  does NOT depend on the I/O parameters, and its computation ultimately requires only the knowledge of the charge status (or voltage level) at each node of the cell.

The wasted energy ( $E_w = E - E_c$ ), on the other hand, does NOT depend on the internal charge status, and can be approximated by a function of the I/O parameters, obtained by fitting the results of electrical simulations.

### 3.2 Advanced Gate Models

For each cell, we provide the layout-extracted internal capacitances and we model the parasitics with constant capacitors to ground, as shown in Fig. 1. We denote by  $\mathcal{N}$  the ordered set of cell nodes ( $n_1, n_2, \dots, n_N$ ), including primary outputs, and by  $C_i$  the capacitance between node  $n_i$  and ground.

In order to compute  $E_c$ , we need to know the voltage level at each node at the beginning and at the end of any transition. Moreover, we need to dynamically determine the set ( $\mathcal{S}$ ) of nodes connected to power supply. To solve these problems, we keep track of the Boolean conditions enabling the connection of each node to ground ( $V_{ss}$ ), to  $V_{dd}$  and to each other node in the cell.

These conditions make up a **connection matrix**  $\mathcal{M}(\mathbf{x})$ , with  $N$  rows and  $N$  columns associated with the cell nodes: entry  $m_{i,j}(\mathbf{x})$  of the matrix is a Boolean function of the cell inputs, taking value 1 for those input configurations for which a conductive path exists between nodes  $n_i$  and  $n_j$ . Two additional columns (namely  $N+1$  and  $N+2$ ) are used to represent the connectivity to power supply ( $d$ ) and ground ( $s$ ).

**Example 3** For the OR gate of Fig. 1, the elements of the first row of the connection matrix are:  $m_{1,1}(\mathbf{x}) = 1$ ,  $m_{1,2}(\mathbf{x}) = x_2'$ ,  $m_{1,3}(\mathbf{x}) = x_2'x_3'$ ,  $m_{1,4}(\mathbf{x}) = 0$ ,  $m_{1,d}(\mathbf{x}) = x_1'$ , and  $m_{1,s}(\mathbf{x}) = x_1(x_2 + x_3)$ . The output node is denoted by  $n_4$ .  $\square$

The efficient handling of the connection matrix is obtained by using Reduced Ordered Binary Decision Diagrams (BDDs) to represent Boolean functions [14]. The square sub-matrix consisting of the first  $N$  columns of  $\mathcal{M}(\mathbf{x})$  is symmetrical, and the BDD-based representation allows a consistent amount of sharing among its entries. Notice that  $\mathcal{M}(\mathbf{x})$  is constructed once for all, during the characterization phase. At run-time, for each input pattern  $\mathbf{x}$  the connection status is then obtained from  $\mathcal{M}(\mathbf{x})$  in linear time, by simple BDD evaluations.

### 3.3 Evaluating $E_c$

During logic simulation, the connection matrix is used both to compute  $E_c$  and to update the charge status. In particular,  $\Delta Q$  can be easily evaluated using the column of  $\mathcal{M}$  associated with  $V_{dd}$ . For a generic cell with  $N$  nodes (including primary outputs), Equation (2) can be rewritten as:

$$\Delta Q = \sum_{i=1}^N m_{i,d}(\mathbf{x}^f) C_i (V_i^f - V_i^i). \quad (3)$$

**Example 4** With respect to Fig. 1, the Boolean conditions enabling the connection of each node of the OR gate to power supply are:  $m_{1,d}(\mathbf{x}) = x_1'$ ,  $m_{2,d}(\mathbf{x}) = x_1'x_2'$ ,  $m_{3,d}(\mathbf{x}) = x_1'x_2'x_3'$ ,  $m_{4,d}(\mathbf{x}) = x_1 + x_2 + x_3$ . For instance, the charge provided by  $V_{dd}$  when the cell inputs switch to  $\mathbf{x}^f = 011$  is then expressed by:

$$\Delta Q = C_1 (V_1^f - V_1^i) + C_4 (V_4^f - V_4^i),$$

where  $C_4$  corresponds to the output load  $C_L$ .  $\square$

Equation (3) requires the complete knowledge of node voltages at the beginning and at the end of the transition. Node voltages are updated by exploiting the connection matrix:

$$V_i^f = m_{i,d}(\mathbf{x}^f) V_{dd} + m_{i,s}(\mathbf{x}^f) V_{ss} + m_{i,float}(\mathbf{x}^f) \frac{\sum_{j=0}^N m_{i,j}(\mathbf{x}^f) C_j V_j^i}{\sum_{j=0}^N m_{i,j}(\mathbf{x}^f) C_j}, \quad (4)$$

where  $m_{i,float}(\mathbf{x}^f)$  takes value 1 whenever node  $n_i$  is floating:  $m_{i,float} = m'_{i,d} m'_{i,s}$ . In practice,  $m_{i,d}(\mathbf{x})$ ,  $m_{i,s}(\mathbf{x})$  and  $m_{i,float}(\mathbf{x}^f)$  work as mutually exclusive selection functions:

- if  $n_i$  is connected to power supply ( $m_{i,d} = 1$ ), the new value of  $V_i$  is  $V_{dd}$ ;
- if  $n_i$  is connected to ground ( $m_{i,s} = 1$ ), the new value of  $V_i$  is  $V_{ss}$ ;
- if  $n_i$  is floating ( $m_{i,float} = 1$ ), the new value of  $V_i$  is computed by taking into account the charge sharing with other channel-connected nodes.

**Example 5** Consider a transition to  $\mathbf{x}^f = 101$  at the inputs to the OR gate of Fig. 1. At the end of the transition, node  $n_1$  is floating and connected only to  $n_2$ . So, the new value of  $V_1$  is given by the charge sharing between  $n_1$  and  $n_2$ :

$$V_1^f = \frac{C_1 V_1^i + C_2 V_2^i}{C_1 + C_2}. \square$$

Notice that Equation (4) also allows us to take implicitly into account the effect of threshold drop on the voltage levels of internal nodes connected to  $V_{dd}$  ( $V_{ss}$ ) through n-channel (p-channel) transistors. For a generic node (say,  $n_i$ ) this is done simply by replacing the nominal values of  $V_{dd}$  and  $V_{ss}$ , with values obtained from electrical simulations (namely,  $V_{dd_i}$  and  $V_{ss_i}$ ) taking into account transistor threshold drops. The main source of error in our estimation of  $E_c$  is then the constant capacitor to ground model (floating capacitors are modeled as capacitors to ground). The effect of nonlinear time-varying junction capacitances and feedthrough parasitic capacitances are approximatively accounted for in the  $E_w$  component of energy dissipation.

### 3.4 Evaluating $E_w$

The main contribution to  $E_w$  is due to the presence of short circuit currents from power supply to ground. The connection matrix can be used to detect conditions for which there is a transient open path between  $V_{dd}$  and  $V_{ss}$ . In practice, a wasted current is drawn from power supply whenever a node that was connected to  $V_{dd}$  for input vector  $\mathbf{x}^i$  is connected to  $V_{ss}$  for input vector  $\mathbf{x}^f$ , or viceversa. For a generic cell with  $N$  nodes, this condition is expressed by

$$f_w(\mathbf{x}^i, \mathbf{x}^f) = \sum_{i=1}^N [m_{i,d}(\mathbf{x}^i)m_{i,s}(\mathbf{x}^f) + m_{i,s}(\mathbf{x}^i)m_{i,d}(\mathbf{x}^f)].$$

If  $f_w = 0$ , then  $E_w = 0$ ; if  $f_w = 1$ , instead,  $E_w$  depends on fan-in and fan-out conditions, represented by the input slopes  $S_1, \dots, S_n$  and by the output load  $C_L$ . Corresponding to any transition, however, short circuit currents are not influenced by those input (output) parameters associated with input (output) signals that don't change.

Since there are no simple closed-form models for the wasted contribution to power consumption, we approximate  $E_w$  with a first-order function of the I/O parameters:

$$E_w = f_w(c_1 S_1 + \dots + c_n S_n + c_{n+1} f_{out} C_L), \quad (5)$$

where  $f_{out}$  is a Boolean flag taking value 1 corresponding to output transitions ( $f_{out} = out(\mathbf{x}^f) \oplus out(\mathbf{x}^i)$ ), and the input slopes are set to 0 if the corresponding inputs do not change ( $x_i^f = x_i^i \implies S_i = 0$ ). Pattern dependency is thus implicitly accounted for.

Coefficients  $c_1, \dots, c_{n+1}$  are computed with min-square fitting of values obtained by circuit simulation. The number of fitting coefficient is *linear* in the number of inputs and outputs of the cell.

### 3.5 Multiple Transitions

Although the model described above is accurate for perfectly aligned multiple input transitions, this assumption is often violated in practice. In the majority of cases, multiple input transitions are slightly misaligned, possibly by short times (compared to the transition time of the gate). In this case a model that computes the power dissipation observing single input transitions may produce large errors, because it will consider a slightly misaligned multiple transition as a sequence of full transitions.

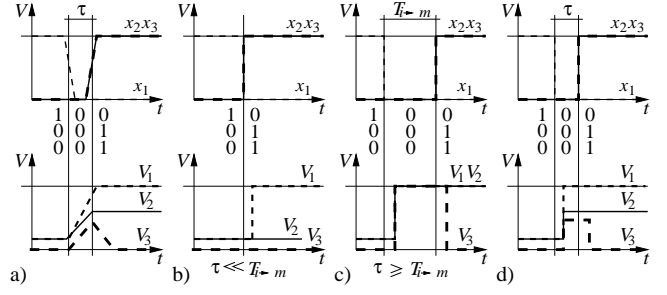


Figure 2: a) Symbolic representation of the effects of a misaligned multiple transition on the internal voltages of the OR gate of Fig. 1. At logic level, we can handle only the two limit situations of b) perfectly aligned input edges, and c) sequences of completely disjoint transitions. However, good estimates of the internal voltages can be obtained in any other case (d) by means of linear interpolation.

Assume that a two input transition from input pattern  $\mathbf{x}^i$  to  $\mathbf{x}^f$  is not perfectly aligned. The misalignment causes an intermediate pattern (say,  $\mathbf{x}^m$ ) to appear at the input of the gate for a short period of time. Assume  $\tau$  to be the delay between the misaligned input transitions (*i.e.*, the input skew). We call **transient time** the time  $T_{i-m}$  needed to reach 90% of the total charge transfer from  $V_{dd}$  to capacitances in the gate (caused by the transition  $\mathbf{x}^i \rightarrow \mathbf{x}^m$ ). There are two limit situations:

- If  $\tau \ll T_{i-m}$ , pattern  $\mathbf{x}^m$  does not appear at the inputs and the energy dissipated is  $E = E_{i-f}$ .
- On the other hand, if  $\tau > T_{i-m}$ , we have two complete transitions and the total energy dissipation is  $E = E_{i-m} + E_{m-f}$ .

We approximate the intermediate cases using a linear interpolation between the two limits. Namely:

$$E = (E_{i-m} + E_{m-f}) \frac{\tau}{T_{i-m}} + E_{i-f} \left(1 - \frac{\tau}{T_{i-m}}\right).$$

Clearly this formula holds if  $\tau < T_{i-m}$ , while if this is not true we have  $E = E_{i-m} + E_{m-f}$ .

**Example 6** Consider the OR gate of Fig. 1, and assume that a multiple input transition occurs from  $\mathbf{x}^i = 100$  to  $\mathbf{x}^f = 011$ , with a misalignment ( $\tau$ ) of  $1ns$  between the falling edge of  $x_1$  and the rising edges of  $x_2$  and  $x_3$ . As shown in Fig. 2.a, the input misalignment gives rise to an intermediate input pattern  $\mathbf{x}^m = 000$ . By electrical simulation, the total energy drawn by the cell is of  $4.15pJ$ .

Since the transient time ( $T_{i-m} = 1.7ns$ ) is greater than the input skew ( $\tau = 1ns$ ), to evaluate  $E$  at logic level we refer to the two limit situations of simultaneous and disjoint input transitions (Fig. 2.b and 2.c), providing:

$E_{i-f} = 0.41pJ$ ,  $E_{i-m} = 4.42pJ$ ,  $E_{m-f} = 3.37pJ$ , respectively. The actual energy estimate is then provided by the following linear interpolation:

$$E = 0.41 \left(1 - \frac{1}{1.7}\right) + (4.42 + 3.37) \frac{1}{1.7} pJ = 4.58pJ,$$

with an error of 4.8% from SPICE.  $\square$

The same approach is used to approximate the charge status of the cell at the end of slightly misaligned multiple transitions, as shown in Fig. 2.d.

The linear approximation is obviously exact at the boundaries, but its accuracy depends on the definition of  $T_{i \rightarrow m}$ . In general,  $T_{i \rightarrow m}$  is strongly pattern dependent and it is not equal to the delay used for event propagation. This is shown in the next section.

### 3.6 Timing Information

As mentioned in previous sections, the power drawn by a cell depends not only on the input patterns applied, but also on signal waveforms and arrival times. At the logic level, however, signal slopes are neither represented nor propagated, and simple delay models (such as unit delay) are used for scheduling the events. These approximations have a critical impact on the accuracy of power estimates.

To solve this problem, we provide accurate models of the three main parameters representing the time behavior of each library-cell:

- the propagation delay  $D$  (used by the simulator for the event scheduling),
- the output slope  $S_{out}$  (used for power estimation of the driven gates),
- the transient time  $T$  (used to handle misaligned multiple transitions).

We approximate these parameters with pattern-dependent functions of both the output load ( $C_L$ ) and the average input slope ( $S_{avg}$ ):

$$\begin{aligned} D &= c_{0d}(\mathbf{x}) + c_{1d}(\mathbf{x})S_{avg} + c_{2d}(\mathbf{x})C_L; \\ S_{out} &= c_{0s}(\mathbf{x}) + c_{1s}(\mathbf{x})S_{avg} + c_{2s}(\mathbf{x})C_L; \\ T &= c_{0t}(\mathbf{x}) + c_{1t}(\mathbf{x})S_{avg} + c_{2t}(\mathbf{x})C_L. \end{aligned} \quad (6)$$

The pattern dependent coefficients  $c$  are determined by means of electrical simulations during the library characterization phase: for each final input configuration  $\mathbf{x}^f$  a set of min-square fitting coefficients is obtained.

In principle, we obtain  $2^n$  different values for each coefficient. In practice, however, the majority of them do not change within large sets of input configurations. Hence, the number of possible assignments of the fitting coefficients is usually small and their pattern dependence can be efficiently modeled by means of ADDs [15].

## 4 Implementation

Routines for both the construction of the connection matrix and the min-square fitting of electrical simulations have been implemented in C using standard packages for BDD and matrix manipulations. VERILOG-XL has been used as simulation platform. The pre-characterized symbolic models of library cells have been written as C functions and made available from logic simulation using the Programming Language Interface (PLI) of VERILOG-XL.

Our simulator has become the first building block of PPP, a new fully integrated synthesis and simulation environment for low-power CMOS circuits. PPP exploits a powerful Web-based interface to address several EDA issues: modularity, platform independence, remote accessibility and user interface standardization. Users can access PPP through the

Internet using their own Web-browsers. The graphical interface of PPP appears as a tree of highly interactive HTML-pages allowing the user to run synthesis and simulation tools on their own circuits, issue specific commands and analyze (partial) results. No software installation is required on the client-side. Users' commands are interpreted by the PPP kernel, and sent to the CAD tools (possibly provided by different developers and distributed on different machines). Partial results are translated and made available from the Web by means of hyperlinks.

Both hardware and software configuration problems are overcome by the *remote-execution* protocol and all the details of tool communication are hidden to the user.

## 5 Experimental Results

We have tested our simulator using a low-power CMOS library (including complex gates and two-level cells). Each library cell has been characterized as described in Section 2, using HSPICE to perform electrical simulations.

To verify the single-pattern accuracy of models provided by the characterization process, we have tested each library cell for all possible test-pairs and for a wide range of fanin and fanout conditions. In the worst case, the average absolute error from HSPICE has been of 4% of the average estimated power, with a standard deviation of 0.2%. The same accuracy has been obtained by applying to each cell a sequence of 100 randomly generated test vectors with 50% of misaligned multiple input transitions.

At last, we have simulated several circuits obtained by mapping a large set of well known benchmark circuits on our test library. Circuits have been simulated by applying randomly generated sequences of 100 input patterns with a clock period of 20ns. Experimental results are reported in Table (1): the worst case error from HSPICE has been of 4.8% on the power consumption of circuit C432. As concerns performance, the CPU times taken on a Sun SPARCstation IPX show that PPP is up to three order of magnitude faster than HSPICE. On the other hand, the performance loss with respect to simple VERILOG simulations based on transition counts with unit delay models is within a factor of 6.

For circuit C17, local power estimates are also reported. The average power drawn by each internal gate has been estimated with a worst case accuracy of 6% from HSPICE. Further results are available in [1, 2].

## 6 Conclusions and Future Work

In this work we presented a cell characterization and logic simulation algorithm that reduces the gap between the accuracy of electrical and logic-level power estimation for cell-based designs. Statistical uncertainties on device characteristics or inaccuracies on wiring capacitance extraction and modeling may lead to mismatches between circuit simulation and measured values that are larger than the inaccuracy of our simulator. As a consequence, our simulation technique gives the designer a level of confidence on power dissipation estimates comparable to those attainable with computationally expensive circuit simulation. The high local accuracy

benchmark		HSPICE		PPP		error %
name	gates	P(mW)	CPU(s)	P(mW)	CPU(s)	
C17	6	<b>0.435</b>	62.4	<b>0.432</b>	0.5	0.7
C432	215	<b>11.510</b>	3120.0	<b>10.954</b>	13.0	4.8
C880	342	<b>16.262</b>	5781.8	<b>16.405</b>	21.9	0.9
C1355	584	<b>23.689</b>	8179.6	<b>24.101</b>	36.4	1.7
C1908	602	<b>33.018</b>	11698.3	<b>33.920</b>	41.6	2.7
C3540	1165	–	–	<b>71.250</b>	118.2	–
C7552	2892	–	–	<b>222.823</b>	272.7	–
cmb	50	<b>1.305</b>	444.9	<b>1.329</b>	2.7	1.8
parity	75	<b>2.381</b>	591.4	<b>2.302</b>	3.6	3.4
count	113	<b>4.985</b>	1290.9	<b>5.081</b>	5.9	1.9
frg1	124	<b>4.934</b>	1316.6	<b>4.957</b>	7.5	0.5
comp	163	<b>6.700</b>	1698.6	<b>6.832</b>	9.6	1.9
x1	345	<b>13.660</b>	4439.7	<b>13.336</b>	20.5	2.4
x4	487	<b>26.698</b>	4445.2	<b>25.549</b>	26.0	4.3
alu2	359	<b>17.881</b>	6863.4	<b>18.550</b>	24.5	3.6
alu4	716	<b>31.565</b>	29685.4	<b>32.878</b>	44.6	4.0
frg2	1253	–	–	<b>57.746</b>	69.1	–
k2	1943	–	–	<b>51.742</b>	73.9	–

Local power consumption of C17

Cell #	Power (mW)		Error	
	HSPICE	PPP	(mW)	(%)
0	0.032	0.031	-0.001	3
1	0.038	0.037	-0.001	3
2	0.043	0.042	-0.001	2
3	0.156	0.154	-0.002	1
4	0.032	0.030	-0.002	6
5	0.134	0.138	+0.004	3
all	0.435	0.432	-0.003	1

Table 1: Experimental results on benchmark circuits. Data refer to sequences of 100 random generated test-vectors with 20ns clock periods (missing results mean that the corresponding simulation exceeded 10 hours of CPU and/or 20 Mbytes of RAM). The table on the right reports the local power estimates provided by PPP and HSPICE for each gate of a NAND-only realization of circuit C17.

makes our tool a valuable source of information for power optimization algorithms that operate locally within a gate-level network.

Our technique achieves better accuracy than previously presented approaches and requires storage of lookup tables with size similar to those required for logic simulation with accurate delay. Important effects such as charge sharing, short circuit current, and misaligned multiple input transitions are taken into account.

To make the interface with pre-existing design flows and RTL simulation completely straightforward we have used VERILOG-XL as simulation platform. Moreover, we have used our simulator as the starting point to develop PPP, an integrated simulation and synthesis EDA tool on the Web.

Future extension of our work will be focused on developing algorithms for gate-level current waveform simulation and hierarchical power analysis.

## Acknowledgements

This work was partially supported by NSF (under contract MIP-9421129) and by AEI (under grant De Castro). We especially would like to thank Prof. Giovanni De Micheli at Stanford for many useful suggestions.

## References

- [1] "DAC96-29.1 support," <http://akebono.stanford.edu/users/cad/publications/DAC96-29.1.html>
- [2] A. Bogliolo, L. Benini, G. De Micheli and B. Riccò, "PPP: A Gate-Level Power Simulator - A World Wide Web Application," *Stanford Technical Report No. CSL-TR-96-691*, 1996.
- [3] C. Huang et al., "The design and implementation of PowerMill," *Proc. of the Int'l Workshop on Low-Power Design*, pp. 105-110, 1995.
- [4] F. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE transaction on VLSI systems*, vol. 2, no. 4, pp. 446-455, 1994.
- [5] C. Y. Tsui et al., "Efficient Estimation of Dynamic Power Dissipation under a Real Delay Model," *Proc. of the IEEE Int'l Conf. on Computer Aided Design*, pp. 224-228, 1993.
- [6] R. Marculescu et al., "Logic level power estimation considering spatiotemporal correlations," *Proc. of the IEEE Int'l Conf. on Computer Aided Design*, pp. 294-299, 1994.
- [7] D. Liu et al., "Power consumption estimation in CMOS VLSI chips," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 6, pp. 663-670, 1994.
- [8] P. Landman et al., "Architectural power analysis, the Dual Bit Type method," *IEEE transaction on VLSI systems*, vol. 3, no. 2, pp. 173-187, 1995.
- [9] R. San Martin et al., "Power-Profiler: optimizing ASICs power consumption at the behavioral level," *Proc. of the Design Automation Conference*, pp. 42-47, 1995.
- [10] L. Benini et al., "Analysis of hazard contribution to power dissipation in CMOS IC's," *Proc. of the Int'l Workshop on Low-Power Design*, pp. 27-32, 1994.
- [11] B. J. George et al., "Power analysis and characterization for semi-custom desing," in *Proc. of the Int'l Workshop on Low-Power Design*, pp. 215-218, 1994.
- [12] J.-Y. Lin et al., "A cell-based power estimation in CMOS combinational circuits," *Proc. of the Int'l Conf. on Computer-Aided Design*, pp. 304-309, 1994.
- [13] H. Sarin et al., "A power modeling and characterization method for logic simulation," *IEEE Custom Integrated Circuits Conference*, pp. 363-366, 1995.
- [14] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, vol. 35, no. 8, pp. 677-691, Aug. 1986.
- [15] R. Bahar et al., "Algebraic decision diagrams and their applications," *Proc. of the Int'l Conf. on Computer-Aided Design*, pp. 188-191, 1993.