Improved Estimation of the Switching Activity for Reliability Prediction in VLSI Circuits

Farid N. Najm

Coordinated Science Lab. & ECE Dept.
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
E-mail: najm@uiuc.edu

Abstract

Estimating the reliability of integrated circuits is a major concern of the semiconductor industry. In CMOS circuits, the extent of node switching activity, called the transition density [1], is a good measure of susceptibility to a variety of reliability problems. However, the density computation algorithm in [1] does not take into account the effect of the inertial delay of logic gates. Thus, the transition density may be severely overestimated in high frequency applications. To overcome this problem, we model the effect of gate delay in the form of a conceptual low-pass filter block that does not allow unacceptably short logic pulses to propagate through. Using a stochastic model of logic signals, we then derive the equations required to propagate the transition density through the filter. We will present experimental results that illustrate the validity and importance of these results.

1. Introduction

The dramatic decrease in feature size and the corresponding increase in the number of devices on a chip have made reliability one of the major concerns in VLSI circuits and systems design. A key observation is that the reliability of a CMOS chip is directly related to the extent of its *switching activity*, i.e., the rate at which its nodes are switching. Indeed, it can be shown that both electromigration and hot-carrier degradation, as well as power dissipation, are aggravated by higher switching frequencies. In general, less active circuits experience lower average currents, both in their transistors and metal lines, and are therefore more reliable.

However, estimating the level of activity is not straightforward because it depends on the specific signals being applied to the circuit primary inputs. These signals are generally unknown during the design phase because they depend on the system in which the chip will eventually be used. A brute-force solution might be to simulate a circuit for all possible inputs. This, however, becomes prohibitively expensive for all but the smallest circuits. To overcome these limitations, the *transition density* was introduced in [1] as a compact measure of switching activity in digital circuits. Simply put, the transition density at a node is the average number of transitions per second at that node, and it can be efficiently evaluated [1] without requiring exact information about the primary input signals.

However, the algorithm for computing the density in [1] is very basic and does not take into account the effect of *inertial delays* of logic gates. Thus, the transition density may be severely overestimated for high speed circuits, as we will now demonstrate.

The density propagation algorithm of [1] can be stated as follows. If y is a Boolean function of $x_1, x_2, \ldots, x_n$, then the densities at x_i, $D(x_i)$ can be used to compute the density at y as follows :

$$D(y) = \sum_{i=1}^{n} P\left(\frac{\partial y}{\partial x_i}\right) D(x_i) \tag{1.1}$$

where the $P(\partial y/\partial x_i)$ coefficients are the probabilities of the Boolean difference terms. The resulting algorithm requires only two pieces of information about every primary input node, namely its *equilibrium probability* $P(x)$ (fraction of time that it is high) and its *transition density* $D(x)$. Thus (1.1) provides a very efficient way of *propagating* these values throughout the circuit so that $D(x)$ is computed for all internal nodes.

The problem with this approach is that it places no checks or restrictions on the *maximum* density (equivalently, the minimum pulse width) at the output node of a Boolean module. This is a result of the simplified timing model. To illustrate, consider an n-input OR gate whose inputs have equal probabilities $P = 0.5$ and equal densities $D = d$. Since the Boolean difference $\frac{\partial y}{\partial x_i}$ is the OR of the $(n-1)$ other inputs, its probability is at least 0.5, which leads to $D(y) \geq (nd/2)$. Thus, for large enough n, the gate output will carry arbitrarily high density, and therefore unrealistically short pulses.

In practice, such short pulses are *not* generated; they are *glitches* that are *filtered out* because the module is not fast enough to respond to them. In order to model this *filtration* effect of the circuit inertial delays, we introduce a new delay block called a *filter block* at every module output. A *filter* block is a delay block with a *low-pass filtering* property, that may be defined as follows : a $0 \to 1$ $(1 \to 0)$ transition at the filter input is transmitted to its output after a delay of τ_1 (τ_0) if and only if its input does not change state during that time. Thus the filter block effectively sets a *minimum pulse width* at the output y : the minimum high (low) pulse width at the filter output is τ_0 (τ_1).

In this paper, we will analyze a filter block and show how to compute the density at its output from that at its input. The basic definitions and main result are given in section 2. Sections 3 and 4 are devoted to experimental results and conclusions. For lack of space, theoretical results will be stated without proof. The reader is referred to [3] for an expanded version of this paper that includes formal proofs of these results.

2. Filter Block Analysis

Let $x(t)$, $t \in (-\infty, +\infty)$, be a logic signal, i.e., it only takes the values 0 or 1. Let $n_x(T)$ be the number of transitions of $x(t)$ in the interval $(\frac{-T}{2}, \frac{+T}{2}]$. The *equilibrium probability* $P(x)$ and *transition density* $D(x)$ were defined in [1] as :

$$P(x) = \lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t)dt \qquad (2.1a)$$

$$D(x) = \lim_{T\to\infty} \frac{n_x(T)}{T} \qquad (2.1b)$$

Based on this, an algorithm was proposed in [1] to propagate $P(x)$ and $D(x)$ throughout the circuit.

A signal $x(t)$ is composed of an alternating sequence of high (corresponding to $x = 1$) and low $(x = 0)$ pulses. We define μ_1 (μ_0) as the average high (low) pulse-width of $x(t)$. Also, let the population of high (low) pulse-widths have the *cumulative distribution function* (cdf) $F_1(t)$ $(F_0(t))$ and the *probability density function* (pdf) $f_1(t) = \frac{d}{dt}F_1(t)$ $(f_0(t) = \frac{d}{dt}F_0(t))$. Thus $F_1(t)$ $(F_0(t))$ is the fraction of high (low) pulses that are shorter than or equal to t. Based on this, define :

$$\hat{F}_0(t) = \int_0^t \frac{z}{\mu_0} f_0(z)dz, \quad \hat{F}_1(t) = \int_0^t \frac{z}{\mu_1} f_1(z)dz$$

In practice, it is reasonable to assume for a general logic signal that two pulses that are sufficiently separated in time will be uncorrelated or independent. We extend this intuitive notion and make the following simplifying assumption which will make possible the solution of a filter block : *The width of every pulse of $x(t)$ is assumed to be independent of all other pulses of $x(t)$.* This assumption is *mild* in the sense that it is approximately true in practice, such as when two pulses are widely separated in time. The main result of this paper is the following theorem that shows how $P(y)$ and $D(y)$ at the output of a filter block can be computed from $P(x)$ and $D(x)$ at its input :

Theorem 1. *For a filter with input x and output y, and given the basic assumption made above, we have :*

$$\begin{aligned} P(y) = P(x) &- \left\{\hat{F}_1(\tau_1) + \frac{\tau_1}{\mu_1}[1 - F_1(\tau_1)]\right\} \frac{[1 - F_0(\tau_0)]}{[1 - F_0(\tau_0)F_1(\tau_1)]} P(x) \\ &+ \left\{\hat{F}_0(\tau_0) + \frac{\tau_0}{\mu_0}[1 - F_0(\tau_0)]\right\} \frac{[1 - F_1(\tau_1)]}{[1 - F_0(\tau_0)F_1(\tau_1)]}[1 - P(x)] \end{aligned} \qquad (2.2a)$$

and :

$$D(y) = \frac{[1 - F_0(\tau_0)][1 - F_1(\tau_1)]}{1 - F_0(\tau_0)F_1(\tau_1)} D(x) \qquad (2.2b)$$

3. Experimental Results and Discussion

Given the equilibrium probability $P(x)$ and transition density $D(x)$ at the primary inputs of a combinational logic circuit, the corresponding values inside the circuit can be computed using [1]. The density values can be used to estimate the susceptibility to reliability problems such as hot-carrier degradation and electromigration, as well as a measure of power dissipation. The density propagation algorithm based on (1.1) was implemented in the program *densim* [1]. The results of this paper have been incorporated into a new implementation of densim by simply applying the filter operation to the output of every gate. The filter parameters τ_0 and τ_1 can be set by the user in the module library; otherwise, they are derived from the propagation delay and rise/fall times.

In order to use the filter equations (2.2), however, we need to know the pulse width distribution functions $F_1(t)$ and $F_2(t)$. The *form* of these distributions is generally unknown, but one can make reasonable assumptions about them, as follows. We will again make use of the intuitive property that the values of a logic signal at widely separated time points are relatively independent. If we extend this property to the point that the future value of a signal is independent of its past, once its present value is specified, then the signal is said to be Markov [2] and its high and low pulses are known to be *exponentially distributed.* In the absence of any other information, therefore, we will assume an exponential distribution.

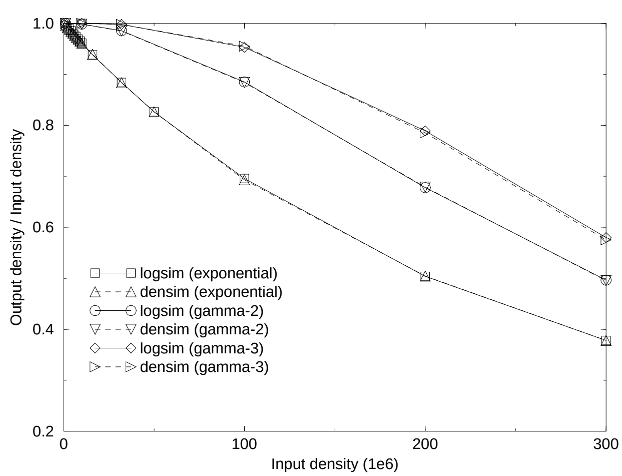

Figure 1. Filter transfer characteristics with $\tau_0 = \tau_1 = 1$ nsec and $P(x) = 0.5$, for different input distributions.

It is prudent at this point to experimentally validate the results of theorem 1. To do this, we applied a randomly generated logic signal to the input of a filter block, and processed the signal as one would in a logic simulator. We then monitored the signal at the filter output. Averaging over a long enough simulation time, the output probability and density should converge to those predicted by theorem 1. This behavior was indeed observed, for three different distributions, as shown in Figs. 1 and 2. The two other distributions marked "gamma-2" and "gamma-3" are closely related to the exponential distribution. In both figures (and in the remainder of this section), the results of logic simulation are marked "logsim," while the results of applying theorem 1 are marked "densim."

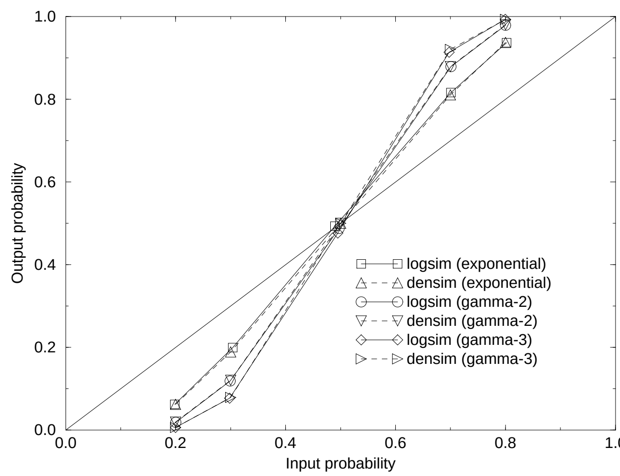

Figure 2. Filter transfer characteristics with $\tau_0 = \tau_1 = 1$ nsec and $D(x) = 1.2e9$, for different input distributions.

The average power dissipation of a circuit is equal to a weighted sum of its node transition densities : $P_{avg} = \sum_{i=1}^{n} \frac{1}{2} C_i V_{dd}^2 D(x_i)$. The plot shown in Fig. 3 compares the average power dissipation, as measured by logic simulation, to that measured by densim with and without the filter. This gives a measure of the overall accuracy in the density computation. The horizontal axis shows the *average frequency* of the signals applied to the circuit primary inputs (the transition density is twice the average frequency). It clearly shows the need for the filter mechanism at higher frequencies. Fig. 4 shows the results of a similar analysis for a 4-bit parallel multiplier and a 4-bit alu.

The above experimental results demonstrate the validity of the results in theorem 1, and the fact that if the filter mechanism is not used, then the basic density propagation algorithm (1.1) will severely deviate from the correct results at higher frequencies.

As a final note, we should say that the improved accuracy afforded by the filter mechanism is obtained at virtually no speed penalty. Equations (2.2) have to be evaluated only once for a logic gate. Thus, the density propagation algorithm remains as efficient as was shown in [1].

On the other hand, the overall approach still has some accuracy problems, even at low frequencies, due to the independence assumptions implicit in (1.1). As was discussed in [1], this is due to node correlations resulting from reconvergent fanout. This issue is part of our continuing work in this area.

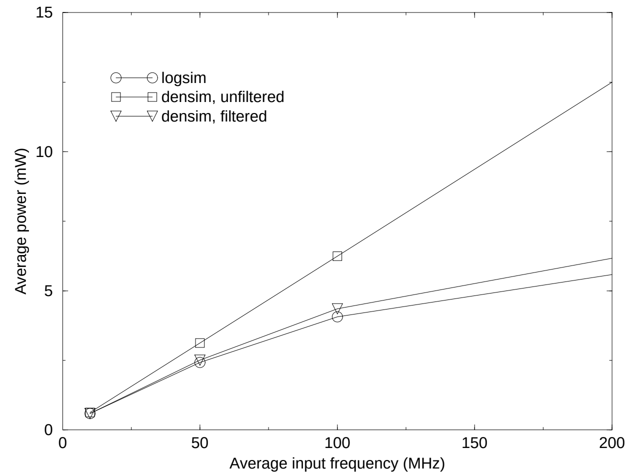

Figure 3. Results for a 32-bit ripple adder with $P(x) = 0.5$ for all inputs, for a wide range of input densities.

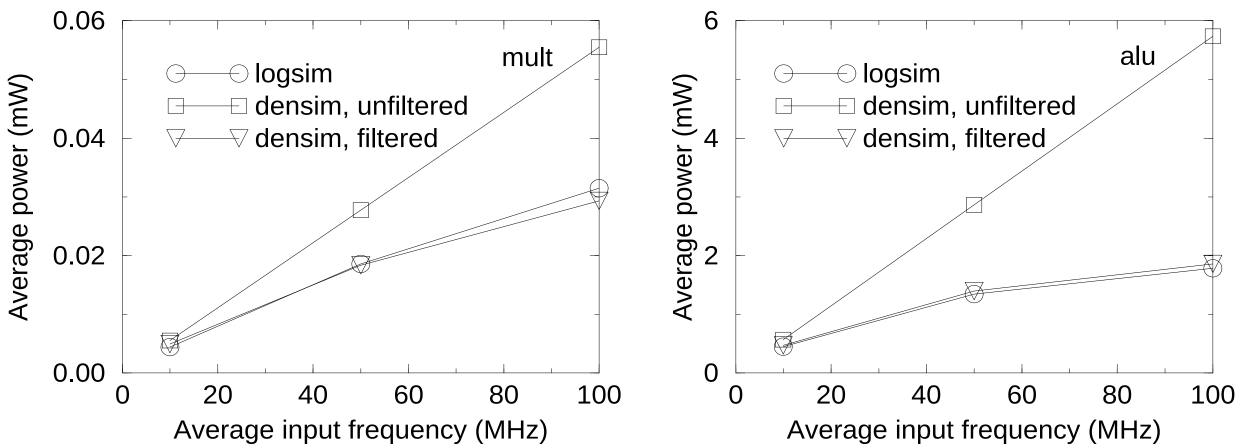

Figure 4. Results for a 4-bit parallel multiplier and a 4-bit alu with $P(x) = 0.5$ for all inputs, for a wide range of input densities.

4. Summary and Conclusions

We have presented an extension to the transition density approach in [1] by taking into account the effect of the inertial delay of a logic gate. In the framework of the stochastic representation of logic signals of [1], we have modeled this effect with a conceptual low-pass filter block. Detailed analysis of this block has yielded compact expressions for the transition density at its output given the density at its input.

Experimental results demonstrate that the filter module behaves as it should, and that the filter mechanism is required in order to maintain accuracy at higher frequencies.

References

[1] F. Najm, "Transition density: a new measure of activity in digital circuits," *IEEE Transactions on Computer-Aided Design*, vol. 12, no. 2, pp. 310-323, February 1993.

[2] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd Edition. New York, NY: McGraw-Hill Book Co., 1984.

[3] F. Najm, "Low-pass Filter for Computing the Transition Density in Digital Circuits," *Submitted to IEEE Transactions on Computer-Aided Design*, 1994.