Transition Density, A New Measure of Activity in Digital Circuits

Farid N. Najm

Semiconductor Process & Design Center
Texas Instruments Inc., MS 369
Dallas, Texas 75265

Abstract

Reliability assessment is an important part of the design process of digital integrated circuits. We observe that a common thread that runs through most causes of run-time failure is the extent of circuit *activity*, i.e., the rate at which its nodes are switching. We propose a new measure of activity, called the *transition density*, which may be defined as the "average switching rate" at a circuit node. Based on a stochastic model of logic signals, we also present an algorithm to propagate density values from the primary inputs to internal and output nodes. To illustrate the practical significance of this work, we demonstrate how the density values at internal nodes can be used to study circuit reliability by estimating (1) the average power & ground currents, (2) the average power dissipation, (3) the susceptibility to electromigration failures, and (4) the extent of hot-electron degradation. The density propagation algorithm has been implemented in a prototype *density simulator*. Using this, we present experimental results to assess the validity and feasibility of the approach. In order to obtain the same circuit activity information by traditional means, the circuit would need to be simulated for thousands of input transitions. Thus this approach is very efficient and makes possible the analysis of VLSI circuits, which are traditionally too big to simulate for long input sequences.

Submitted to the IEEE Transactions on Computer-Aided Design, 1991.

1. Introduction

A major portion of the design time of digital integrated circuits is dedicated to functional verification and reliability assessment. Of these two, reliability assessment is a more recent problem whose severity has steadily increased in proportion to chip density. As a result, CAD tools that evaluate the susceptibility of a design to run-time failures are becoming increasingly important.

Chip run-time failures can occur due to a variety of reasons, such as excessive power dissipation, electromigration, hot-electron degradation, voltage drop, aging, and others. In CMOS logic circuits, the rate at which node transitions occur is a good indicator of the circuit's susceptibility to run-time failures. For example, both power dissipation and electromigration in the power lines are directly related to the power supply current which, in CMOS, is non-zero only during transitions. Hot-electron degradation is related to the MOSFETs substrate current, which, for CMOS, is also only significant during transitions. Thus, the rate at which node transitions occur, i.e., the extent of circuit *activity*, may be thought of as a measure of a failure-causing *stress.* However, there has traditionally been no way of *quantifying* this activity because logic signals are in general non-periodic and, thus, have no fixed switching *frequency.*

This paper proposes a novel measure of activity that we call the *transition density*, along with a simulation technique to compute the density at every circuit node. The transition density may be defined as the "average switching rate," a more rigorous definition will be given in section 2. Preliminary results of this work have appeared in [1].

To further motivate the notion of transition density, consider the problem of estimating the average power drawn by a CMOS gate. If the gate has output capacitance C and generates a simple clock signal with frequency f, then the average power dissipated is $CV_{dd}^2 f$, where V_{dd} is the power supply voltage. In general, since logic signals may not be periodic, the notion of frequency cannot be used. Instead, one may compute the power as follows. If $x(t)$ is the logic signal at the gate output and $n_x(T)$ is the number of transitions of $x(t)$ in the time interval $(\frac{-T}{2}, \frac{+T}{2}]$, then the average power is :

$$P_{av} = \lim_{T\to\infty} V_{dd} \frac{C V_{dd} n_x(T)/2}{T} = \frac{1}{2} C V_{dd}^2 \left\{ \lim_{T\to\infty} \frac{n_x(T)}{T} \right\} \tag{1.1}$$

In the next section we define the transition density to be the last (limit) term in (1.1).

Naturally, one can approximate $\lim_{T\to\infty} n_x(T)/T$ by simulating the circuit for a "large enough" number of input transitions while monitoring $n_x(T)$ at every node. The ambiguity in the phrase "large enough" is precisely the problem with this traditional approach. It is impossible to determine a priori how long the simulation should be. Furthermore, long simulations of large circuits are very expensive. However, we will show that, if the transition densities at the circuit primary inputs are given, they can be efficiently propagated into the circuit to give the transition density at every internal and output node. In other words, we use the *limits* $\lim_{T\to\infty} n_x(T)/T$ at the circuit inputs to *directly* compute the corresponding *limits* inside the circuit.

The propagation algorithm involves a single-pass over the circuit and computes the transition densities at all the nodes. It may be thought of as a *simulation* of the circuit in which one studies the density of its internal signals that correspond to input signals with specified densities; it has been implemented in a prototype *density simulator*, called DENSIM. In order to obtain the same circuit activity information by traditional means, the circuit must be simulated for thousands of input transitions. Thus this approach is very efficient and makes possible the analysis of VLSI circuits, which are traditionally too big to simulate for long input sequences.

It turns out to be highly beneficial, in terms of the theoretical results to be presented, to cast the problem in a stochastic (probability theory) setting. Thus, in the following two sections, we will start with definitions of "idealized logic signals," and then present a stochastic model of logic signals that is essential to the density propagation theorem. Based on these concepts, we then show in section 4 how the transition density can be efficiently propagated from inputs to outputs. In section 5, we study a number of practical applications of the density concept. Namely, we demonstrate how the density values at internal nodes can be used to estimate (1) The average power & ground currents, (2) The average power dissipation, (3) The susceptibility to electromigration failures, and (4) The extent of hot-electron degradation. Experimental results are presented in section 6, and section 7 contains a summary and conclusions.

Appendix A presents the existence proofs of the equilibrium probability and transition density, and appendix B presents a new application for Binary Decision Diagrams (BDDs) in computing the *probability* of a Boolean function.

2. Ideal Logic Signals

Let $x(t), t \in (-\infty, +\infty)$, be a function of time that takes the values 0 or 1. We use such time functions to model *logic signals* in digital circuits. This ideal model neglects waveform details such as the rise/fall times, glitches, over/under-shoots, etc.

Definition 1. *The equilibrium probability of $x(t)$, to be denoted by $P(x)$, is defined as :*

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

The reason for the name "equilibrium probability" will become clear later on. It is easy to observe, however, that $P(x)$ is the fraction of time that $x(t)$ is in the 1 state. It is also the average value of $x(t)$, over all time. Thus, for instance, a 25% duty cycle clock signal, i.e., one that is high for 1/4th of its period, has $P(x) = 0.25$. The following proposition guarantees that the equilibrium probability is always well-defined.

Proposition 1. *For a logic signal $x(t)$, the limit in (2.1) always exists.*

Proof : See appendix A.

The discontinuity points of $x(t)$ represent *transitions* in the logic signal. Let $n_x(T)$ be the number of transitions of $x(t)$ in the time interval $(-\frac{T}{2}, +\frac{T}{2}]$.

Definition 2. *The transition density of a logic signal $x(t), t \in (-\infty, +\infty)$, is defined as :*

$$D(x) \triangleq \lim_{T\to\infty} \frac{n_x(T)}{T} \tag{2.2}$$

The reason for the name "transition density" will become clear later on. It should be clear, however, that $D(x)$ is the average number of transitions per unit time. Thus, a 10MHz clock signal has $D(x) = 20 \times 10^6$. The power of the $P(x)$ and $D(x)$ concepts is that they apply equally well to both periodic (clock) and non-periodic signals. In the remainder of this section, we study the existence of the limit in (2.2).

The time between two consecutive transitions of $x(t)$ will be referred to as an *inter-transition time*. Let μ be the average value of all the inter-transition times of $x(t)$. Likewise, let μ_1 (μ_0) be the average of the high (low), i.e., corresponding to $x(t) = 1$ (0), inter-transition times of $x(t)$. It should be clear that $\mu = \frac{1}{2}(\mu_0 + \mu_1)$. In general, there is no guarantee of the existence of μ, μ_0, and μ_1. If the number of transitions in positive time is *finite*, then we say that there is an *infinite* inter-transition time following the last transition, and $\mu = \infty$. A similar convention is made for negative time. We define μ_f to be the average of all the *finite* inter-transition times of $x(t)$. In general, there is also no guarantee of the existence of μ_f.

Proposition 2. *Two parts :*

(i) If μ_f exists and is non-zero, then $D(x)$ exists.

(ii) If μ_0 & μ_1 exist, and $\mu \neq 0$, then $D(x)$ exists and we have :

$$P(x) = \frac{\mu_1}{\mu_0 + \mu_1} \qquad \text{and} \qquad D(x) = \frac{2}{\mu_0 + \mu_1} \tag{2.3a, b}$$

Proof : See appendix A.

In order to guarantee that the density is always well-defined, we make the following basic assumption about every logic signal $x(t)$:

Basic Assumption : *The average finite inter-transition time μ_f exists and is non-zero.*

Logic signals for which this assumption does not hold are considered pathological, and are excluded from the analysis. It can be shown (see appendix A) that another more stringent sufficient condition for the existence of (2.2) is that there be a non-zero lower bound (however small) on the inter-transition times of $x(t)$. This condition is easily satisfied in all practical cases, so that our basic assumption is very mild indeed.

3. The Companion Process of Logic Signals

We will use **bold font** to represent random quantities. We denote the probability of an event A by $\mathcal{P}\{A\}$ and, if $\mathbf{x}$ is a random variable, we denote its mean (or expected value) by $E[\mathbf{x}]$ and its distribution function by $F_{\mathbf{x}}(a) \triangleq \mathcal{P}\{\mathbf{x} \leq a\}$.

Let $\mathbf{x}(t)$, $t \in (-\infty, +\infty)$, be a *stochastic process* [2] that takes the values 0 or 1, transitioning between them at *random* transition times. Such a process is called a *0-1 process* (see [3], pp. 38–39). A logic signal $x(t)$ can be thought of as a *sample* of a 0-1 stochastic process $\mathbf{x}(t)$, i.e., $x(t)$ is one of an infinity of possible signals that comprise the family $\mathbf{x}(t)$.

A stochastic process is said to be *strict-sense stationary* (SSS) if its statistical properties are invariant to a shift of the time origin [2]. Among other things, the mean $E[\mathbf{x}(t)]$ of such a process is a constant, independent of time, and will be denoted by $E[\mathbf{x}]$. It will be shown below that a logic signal is always a sample of a SSS 0-1 process.

Let $\mathbf{n_x}(T)$ denote the (random) number of transitions of $\mathbf{x}(t)$ in $(-\frac{T}{2}, +\frac{T}{2}]$. If $\mathbf{x}(t)$ is SSS, then $E[\mathbf{n_x}(T)]$ depends only on T, and is independent of the location of the time origin.

Proposition 3. *If $\mathbf{x}(t)$ is SSS, then the mean $E[\mathbf{n_x}(T)/T]$ is a constant, independent of T.*

Proof : Let $t_1 < t_2 < t_3$ be three arbitrary points along the time axis. Let $\mathbf{n_1}$ be the number of transitions in $(t_1, t_2]$, $\mathbf{n_2}$ be the number of transitions in $(t_2, t_3]$, and $\mathbf{n_3}$ be the number of transitions in $(t_1, t_3]$. Then $\mathbf{n_3} = \mathbf{n_1} + \mathbf{n_2}$, and $E[\mathbf{n_3}] = E[\mathbf{n_1}] + E[\mathbf{n_2}]$. Let $T_1 = t_2 - t_1$ and $T_2 = t_3 - t_2$. Since $\mathbf{x}(t)$ is SSS, then $E[\mathbf{n_1}] = E[\mathbf{n_x}(T_1)]$, $E[\mathbf{n_2}] = E[\mathbf{n_x}(T_2)]$, and $E[\mathbf{n_3}] = E[\mathbf{n_x}(T_1 + T_2)]$. Hence $E[\mathbf{n_x}(T_1 + T_2)] = E[\mathbf{n_x}(T_1)] + E[\mathbf{n_x}(T_2)]$. Since this is true for arbitrary T_1 and T_2, it means that, in general, $E[\mathbf{n_x}(T)] = kT$, where k is a positive constant, which completes the proof. ■

A constant-mean stochastic process $\mathbf{x}(t)$ is said to be *mean-ergodic* [2] if :

$$\lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} \mathbf{x}(t)dt \overset{1}{=} E[\mathbf{x}] \tag{3.1}$$

where we have used the symbol "$\overset{1}{=}$" to denote *convergence with probability 1.* The reader is referred to [2], pp. 188–191, for a discussion of the different stochastic convergence modes. We reserve the symbol "=" to indicate *convergence everywhere* for random quantities. It will be shown below that a logic signal is always a sample of a SSS mean-ergodic 0-1 process.

Let $\boldsymbol{\tau} \in (-\infty, +\infty)$ be a random variable with the probability distribution function $F_{\boldsymbol{\tau}}(t) = 1/2$ for any finite t, and with $F_{\boldsymbol{\tau}}(-\infty) = 0$ & $F_{\boldsymbol{\tau}}(+\infty) = 1$. If $F_{\boldsymbol{\tau}_T}(t)$ is the uniform distribution over $[\frac{-T}{2}, \frac{+T}{2}]$, then $\lim_{T\to\infty} F_{\boldsymbol{\tau}_T} = F_{\boldsymbol{\tau}}$. Thus, one might say that $\boldsymbol{\tau}$ is uniformly distributed over the whole real line $\mathcal{R}$.

We now use $\boldsymbol{\tau}$ to build from $x(t)$ an important 0-1 process $\mathbf{x}(t)$, defined as follows.

Definition 3. *Given a logic signal $x(t)$ and a random variable $\boldsymbol{\tau}$, uniformly distributed over $\mathcal{R}$, define a 0-1 stochastic process $\mathbf{x}(t)$, called the companion process of $x(t)$, given by :*

$$\mathbf{x}(t) \overset{\triangle}{=} x(t + \boldsymbol{\tau}) \tag{3.2}$$

For any given $t = t_1$, $\mathbf{x}(t_1)$ is the random variable $x(t_1 + \boldsymbol{\tau})$ - a function of $\boldsymbol{\tau}$. Thus the stochastic process $\mathbf{x}(t)$ is well-defined. Intuitively, $\mathbf{x}(t)$ is a family of shifted copies of $x(t)$, each shifted by a value of the random variable $\boldsymbol{\tau}$. Thus, not only is $x(t)$ a sample of $\mathbf{x}(t)$, but we also have the following.

Proposition 4. *If* $\mathbf{x}(t)$ *is the companion process of a logic signal* $x(t)$*, then the following "convergence everywhere" results are true :*

$$\lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} \mathbf{x}(t)dt = P(x) \tag{3.3}$$

$$\lim_{T\to\infty} \frac{\mathbf{n_x}(T)}{T} = D(x) \tag{3.4}$$

Proof : To prove (3.3), we need to show that, for any given finite $\tau_1 \in \mathcal{R}$, the difference $\Delta_P \triangleq \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t+\tau_1)dt - \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t)dt$ tends to zero as $T \to \infty$. This can be written as :

$$\Delta_P = \frac{1}{T}\int_{\frac{-T}{2}+\tau_1}^{\frac{+T}{2}+\tau_1} x(t)dt - \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t)dt = \frac{1}{T}\int_{\frac{+T}{2}}^{\frac{+T}{2}+\tau_1} x(t)dt - \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{-T}{2}+\tau_1} x(t)dt \tag{3.5}$$

Since $x(t) \in \{0,1\}$, then $|\Delta_P| \leq \frac{|\tau_1|}{T}$ must go to 0 as $T \to \infty$.

Likewise, to prove (3.4) we must show that the difference $\Delta_D \triangleq \frac{1}{T}\{$"the number of transitions in $(\frac{-T}{2}+\tau_1, \frac{+T}{2}+\tau_1]$" $-$ "that in $(\frac{-T}{2}, \frac{+T}{2}]$"$\}$, goes to 0 as $T \to \infty$. Notice that :

$$\frac{n_x(T-2|\tau_1|)}{T} - \frac{n_x(T)}{T} \leq \Delta_D \leq \frac{n_x(T+2|\tau_1|)}{T} - \frac{n_x(T)}{T} \tag{3.6}$$

If $x(t)$ has a finite number of transitions, then $\lim_{T\to\infty} n_x(T \pm 2|\tau_1|) = \lim_{T\to\infty} n_x(T) < \infty$, and $\lim_{T\to\infty} \Delta_D = 0$. Otherwise, notice that $1/D(x) = \lim_{T\to\infty}(T \pm 2|\tau_1|)/n_x(T \pm 2|\tau_1|) = \lim_{T\to\infty} T/n_x(T \pm 2|\tau_1|)$. Rewriting this as $\lim_{T\to\infty} n_x(T \pm 2|\tau_1|)/T = D(x)$ shows that Δ_D must go to 0 as $T \to \infty$.

Since this is true for any $\tau_1 \in \mathcal{R}$, then the convergence is *everywhere*, in the sense that *every* value of $\boldsymbol{\tau}$ will lead to convergence. ■

Theorem 1. *The companion process* $\mathbf{x}(t)$ *of a logic signal* $x(t)$ *is SSS and mean-ergodic with* $E[\mathbf{x}] = \mathcal{P}\{\mathbf{x}(t) = 1\} = P(x)$ *and :*

$$E\left[\frac{\mathbf{n_x}(T)}{T}\right] = D(x) \tag{3.7}$$

Proof : At $t = 0$, we have $E[\mathbf{x}(0)] = E[x(\boldsymbol{\tau})]$. An interesting property of $\boldsymbol{\tau}$ is that if a is a constant then $a + \boldsymbol{\tau}$ has the same distribution as $\boldsymbol{\tau}$. Indeed, if $F_{a+\boldsymbol{\tau}}(t)$ is the distribution function of $a + \boldsymbol{\tau}$, then $F_{a+\boldsymbol{\tau}}(t) = \mathcal{P}\{a + \boldsymbol{\tau} \leq t\} = \mathcal{P}\{\boldsymbol{\tau} \leq t - a\} = 1/2 = F_{\boldsymbol{\tau}}(t)$. Therefore, since $t + \boldsymbol{\tau}$ and $\boldsymbol{\tau}$ are identically distributed, we have $E[x(t+\boldsymbol{\tau})] = E[x(\boldsymbol{\tau})]$, which means that $\mathbf{x}(t)$ is a *constant-mean* process with :

$$E[\mathbf{x}(t)] = E[\mathbf{x}(0)] = E[x(\boldsymbol{\tau})], \text{ for any time } t. \tag{3.8}$$

Let $\mathcal{R}_a$ be a subset of the real line $\mathcal{R}$ defined by $\mathcal{R}_a \triangleq \{t \in \mathcal{R} : x(t) = 1, x(t+a) = 1\}$. It is clear that $\mathcal{P}\{x(\boldsymbol{\tau}) = 1, x(\boldsymbol{\tau}+a) = 1\} = \mathcal{P}\{\boldsymbol{\tau} \in \mathcal{R}_a\}$. Likewise, $\mathcal{P}\{x(t+\boldsymbol{\tau}) = 1, x(t+\boldsymbol{\tau}+a) = 1\} = \mathcal{P}\{t+\boldsymbol{\tau} \in \mathcal{R}_a\}$. However, since $\boldsymbol{\tau}$ and $t+\boldsymbol{\tau}$ are identically distributed, the two probabilities $\mathcal{P}\{\boldsymbol{\tau} \in \mathcal{R}_a\}$ and $\mathcal{P}\{t+\boldsymbol{\tau} \in \mathcal{R}_a\}$ must be equal, which leads to :

$$\mathcal{P}\{\mathbf{x}(t) = 1, \mathbf{x}(t+a) = 1\} = \mathcal{P}\{\mathbf{x}(0) = 1, \mathbf{x}(a) = 1\} = \mathcal{P}\{\boldsymbol{\tau} \in \mathcal{R}_a\}, \text{ for any time } t. \tag{3.9}$$

Consequently, the joint distribution of $\mathbf{x}(t)$ and $\mathbf{x}(t+a)$, i.e., $F_{\mathbf{x}(t),\mathbf{x}(t+a)}(x_1, x_2)$, is independent of t, and depends only on a, which makes $\mathbf{x}(t)$ wide-sense stationary [2] (WSS). By extending this argument to $a_1, \ldots, a_n$, it follows that $F_{\mathbf{x}(t),\mathbf{x}(t+a_1),\ldots,\mathbf{x}(t+a_n)}(x_1, \ldots, x_n)$ is independent of t, and $\mathbf{x}(t)$ is strict-sense stationary (SSS).

To prove mean-ergodicity, and in view of (3.3), it suffices to show that $E[\mathbf{x}] = P(x)$. Consider the random variable $\eta_T \triangleq \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{+T}{2}} \mathbf{x}(t)dt$. From (3.3) we have $\lim_{T\to\infty} \eta_T = P(x)$, where this is *convergence everywhere*. Therefore $\lim_{T\to\infty} E[\eta_T] = P(x)$. By linearity of the expected value operator, this can be rewritten :

$$\lim_{T\to\infty} \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{+T}{2}} E[\mathbf{x}(t)]dt = P(x) \tag{3.10}$$

But $E[\mathbf{x}(t)]$ is a constant. Therefore the left hand side of (3.10) is simply $E[\mathbf{x}]$, and mean-ergodicity follows, with $E[\mathbf{x}] = \mathcal{P}\{\mathbf{x}(t) = 1\} = P(x)$.

To complete the proof, we will prove (3.7) by repeating the argument used for η_T. By (3.4), the random variable $\mathbf{n_x}(T)/T$ converges *everywhere* to $D(x)$. Therefore its mean must also converge to $D(x)$. Since, by proposition 3, its mean is a constant, independent of T, then (3.7) follows. ∎

We are now in a position to comment on the names "equilibrium probability" for $P(x)$ and "transition density" for $D(x)$. For a 0-1 process, $\mathcal{P}\{\mathbf{x}(t) = 1\} = E[\mathbf{x}(t)]$. Thus, by (3.3) and since $\mathbf{x}(t)$ is mean-ergodic, $P(x)$ is the constant probability that $\mathbf{x}(t) = 1$. The name "equilibrium probability" is inspired from the special case when the inter-transition times of a 0-1 process $\mathbf{x}(t)$ are independent exponentially distributed random variables. In that case, the process is the well-known two-state continuous-time Markov process (see [2], pp. 392–393) whose state probability *tends to an equilibrium value* for $t \to \infty$, at which time it becomes SSS (see [4], pp. 272–273).

By (3.7), $D(x)$ is the expected "average number of transitions per unit time," which we compactly refer to as "transition density." This name is inspired by the *density* of random Poisson points (see [2], page 58). If a large number of points are chosen on the time axis at random, then the "number of points in a given interval" is a random variable with a Poisson distribution whose *density* parameter λ is the "expected number of points per unit time." The *points* that we are concerned with in this paper are the time points at which transitions occur, but we make no assumption about their distribution. The remark about Poisson points is meant only to motivate the terminology.

4. Density Simulation

A digital circuit provides a mapping from the logic signals at its primary input nodes to those at its internal and output nodes. In the following, we use the term "internal nodes" to refer to the primary output nodes as well as other proper internal circuit nodes.

If we consider the companion process of each such logic signal, the circuit may be seen as mapping stochastic processes at its inputs to similar processes at its internal nodes. The statistics (such as density and probability) of the internal processes are completely determined by those at the primary inputs. In fact, we will demonstrate in this section that the density and probability of internal processes can be efficiently *computed* from those at the primary inputs.

We assume that the primary input processes are *mutually-independent*. Therefore, since these inputs are individually SSS, they are also *jointly* SSS. In terms of the underlying logic signals $x(t)$, this assumption means that the signal values are not *correlated*, so that if one of them is 1, then the average fraction of time that another is 1 (or 0) is unaltered.

Given the density and probability values of the companion processes at the primary inputs, we will present an algorithm to propagate them into the circuit to derive the corresponding values at internal nodes. We consider the circuit to be an interconnection of *logic modules*, each representing a certain (combinational) Boolean function and possessing certain delay characteristics. The propagation of density and probability will then proceed on a per-module basis from primary inputs to primary outputs, a process that we refer to as *density simulation*.

4.1. Propagation through a single module

Consider a multi-input multi-output logic module M, whose outputs are Boolean functions of its inputs, as shown in Fig. 1. M may be a single logic gate or a higher level circuit block. We assume that the inputs to M are mutually-independent companion processes. The validity of this assumption will be discussed in section 4.2.

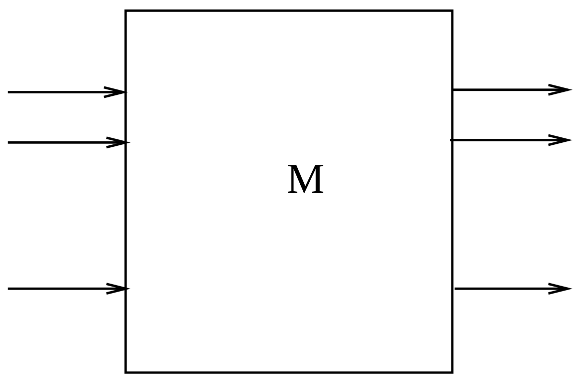

Figure 1. Logic Module M.

We use a *simplified timing model* of circuit behavior, as follows. We assume that an input transition that *does* get transmitted to an output node is delayed by a *propagation delay* time of τ_p. Different propagation delays may be associated with different input-output node pairs. Implicit in this model is the simplifying assumption that the propagation delay is independent of the values at other inputs of M.

In effect, we decouple the delays inside M from its Boolean function description by introducing a special-purpose delay block to model the delays between every pair of input & output nodes, as shown in Fig. 2. The block M' is a zero-delay logic module that implements the same Boolean function as M.

Since the input signals are SSS, then the output of the delay block has the same statistics as its input, and therefore has the same probability and density. As for the zero-delay module M', we now consider the problem of propagating equilibrium probabilities and transition densities from its inputs to its outputs.

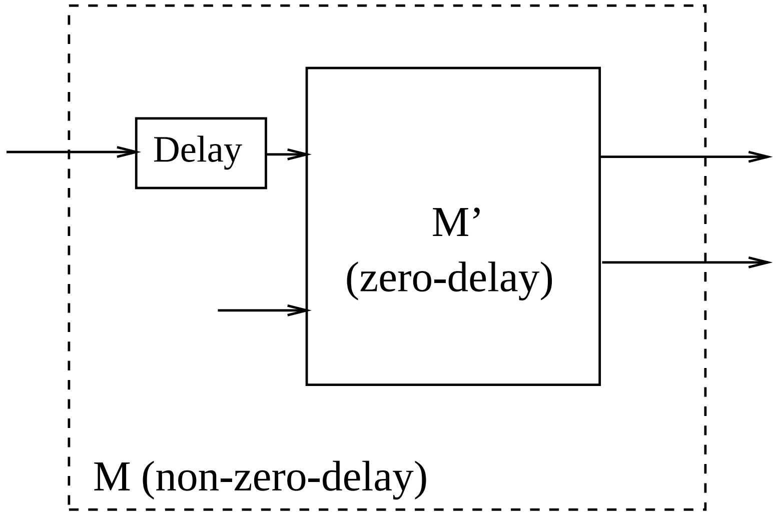

Figure 2. Decoupling of delays.

Since $P(x) = \mathcal{P}\{\mathbf{x}(t) = 1\}$ (by theorem 1) and M' has zero delay, then the problem of propagating equilibrium probabilities through it is identical to that of propagating *signal probabilities* through logic circuits, which has been well-studied [5]–[9]. Since the internal structure of M' is not known, the problem is actually even more generic than that, and can be expressed as "given a Boolean function $f(x_1, \ldots, x_n)$ and that each x_i can be high with probability $P(x_i)$, what is the probability that f is high?" Any number of published techniques can be used to solve this problem. However, we have chosen (for reasons that will become clear below) to investigate a new approach based on Binary Decision Diagrams [10, 11] (BDDs) which have recently become popular in the verification and synthesis areas. Appendix B describes how we use BDDs to compute the probability of a Boolean function.

We consider next the density propagation problem. Recall the concept of *Boolean Difference* : if y is a Boolean function that depends on x, then the Boolean difference of y with respect to x is defined as :

$$\frac{\partial y}{\partial x} \triangleq y|_{x=1} \oplus y|_{x=0} = y(x) \oplus y(\overline{x}) \tag{4.1}$$

where $\oplus$ denotes the exclusive-or operation. Note that, if x is an input and y is an output of M', then $\partial y/\partial x$ is a Boolean function that does *not* depend on x, but may depend on all other inputs of M'. Therefore, $\partial y/\partial x$ and x are independent. A crucial observation is that if $\partial y/\partial x$ is 1, then a transition at x will cause a (simultaneous) transition at y, otherwise *not.* Since the input processes are SSS, then $\partial\mathbf{y}/\partial\mathbf{x}$ is also SSS; in fact it is a companion process with equilibrium probability $P(\partial y/\partial x)$. We are now ready to prove the following :

Theorem 2. *If the inputs $\mathbf{x_i}(t), i = 1, \ldots, n,$ of a zero-delay logic module are independent companion processes with transition densities $D(x_i)$, then the densities at its outputs $\mathbf{y_j}(t), j = 1, \ldots, m$ are given by :*

$$D(y_j) = \sum_{i=1}^{n} P\left(\frac{\partial y_j}{\partial x_i}\right) D(x_i) \tag{4.2}$$

Proof : Let $\mathbf{t_{ik}}, k = 1, 2, \ldots, \mathbf{n_{x_i}}(T)$, be the sequence of transition time points of $\mathbf{x_i}(t)$ in $(\frac{-T}{2}, \frac{+T}{2}]$. Consider the sequence of random variables $\frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(\mathbf{t_{ik}}), k = 1, 2, \ldots, \mathbf{n_{x_i}}(T)$, defined for every input-output pair $(\mathbf{x_i}, \mathbf{y_j})$ of M'.

Since $\frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(t)$ is SSS and independent of $\mathbf{x_i}(t)$, then $\mathcal{P}\left\{\frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(\mathbf{t_{ik}}) = 1\right\} = P\left(\frac{\partial y_j}{\partial x_i}\right)$ is the same for any k. Therefore, $\frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(\mathbf{t_{ik}}), k = 1, 2, \ldots, \mathbf{n_{x_i}}(T)$, is a sequence of identically-distributed (not necessarily independent) random variables, with mean $P\left(\frac{\partial y_j}{\partial x_i}\right)$.

Since $\frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(\mathbf{t_{ik}}) = 1$ if and only if the kth transition of $\mathbf{x_i}(t)$ is transmitted to $\mathbf{y_j}(t)$, then the number of transitions of $\mathbf{y_j}(t)$ in $(\frac{-T}{2}, \frac{+T}{2}]$ is given by :

$$\mathbf{n_{y_j}}(T) = \sum_{i=1}^{n} \sum_{k=1}^{\mathbf{n_{x_i}}(T)} \frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(\mathbf{t_{ik}}) \tag{4.3}$$

Taking the expected value of both sides gives :

$$E\left[\mathbf{n_{y_j}}(T)\right] = \sum_{i=1}^{n} E\left[\sum_{k=1}^{\mathbf{n_{x_i}}(T)} \frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(\mathbf{t_{ik}})\right] \tag{4.4}$$

Since $\frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(t)$ is independent of $\mathbf{x_i}(t)$, and if n is some positive integer, then :

$$E\left[\frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(\mathbf{t_{ik}}) \;\middle|\; \mathbf{n_{x_i}}(T) = n\right] = E\left[\frac{\partial \mathbf{y_j}}{\partial \mathbf{x_i}}(\mathbf{t_{ik}})\right] = P\left(\frac{\partial y_j}{\partial x_i}\right) \tag{4.5}$$

Using [2], p. 183, these facts lead to :

$$E\left[\mathbf{n_{y_j}}(T)\right] = \sum_{i=1}^{n} P\left(\frac{\partial y_j}{\partial x_i}\right) E\left[\mathbf{n_{x_i}}(T)\right] \tag{4.6}$$

which, dividing by T and using (3.7), leads to the required result (4.2). ∎

If the Boolean difference is available, then evaluating $P\left(\frac{\partial y_j}{\partial x_i}\right)$ is no more difficult than evaluating the probability of a Boolean function knowing those of its inputs. Note that if M is a 2-input AND gate, with inputs x_1 & x_2, and output y, then $P\left(\frac{\partial y}{\partial x_1}\right) = P(x_2)$. In more complex situations, the "compose" and "xor" functions of the BDD package [11] can be used to evaluate the Boolean difference using equation (4.1). The BDD-based algorithm given in the appendix (for computing the probability of a Boolean function) can then be used to compute $P\left(\frac{\partial y_j}{\partial x_i}\right)$.

4.2. Global propagation strategy

The assumption was made at the beginning of the previous sub-section that the inputs to a module are *independent*. Even though this is true at the primary inputs (as we have assumed), it may not be true for internal nodes. Circuit topologies that include reconvergent fanout and feedback will cause internal nodes to be correlated, and destroy the independence property. This problem is central to any circuit analysis based on a statistical representation of signals, and can usually be taken care of by using heuristics that trade-off accuracy for speed [5-9].

Based on our previous experience with the propagation of probability waveforms [12], we have found that, if the modules are large enough so that tightly coupled nodes (such as in latches or small cells) are kept inside the same module, then the coupling outside the modules is sufficiently low to justify an independence assumption. While this does take into account the correlations inside a module, it may create inaccuracies because internal delays are lumped together. Furthermore, performance may be sacrificed because the BDDs can become too large. Section 6 will investigate this speed-accuracy trade-off.

5. Practical Applications

Once the density at every internal node has been computed, these values can be used in a post-processing step to investigate various circuit properties. We present here four different applications of the density concept in CMOS circuits.

5.1. Average power/ground bus currents

Consider the problem of computing the average current in the power or ground bus branches. We will consider only the case of the power bus, since that of the ground bus is identical.

A convenient approximation is to view the bus as an interconnection of lumped resistors, with lumped capacitors to ground, i.e., a linear RC network. Some nodes of this network are connected to the external V_{dd} power supply, while others (referred to as *contacts*) are connected to the various circuit components, e.g. CMOS gates, drawing power supply current. Let $i_k(t), k = 1, 2, \ldots, n$, be the various current waveforms that the circuit draws at these contact nodes. Let $i_j(t), j = 1, 2, \ldots, m$, be the various current waveforms in the bus branches. The bus can now be viewed as a *linear time-invariant* (LTI) system whose outputs $i_j(t)$ are related to its inputs $i_k(t)$ by the convolutions :

$$i_j(t) = \sum_{k=1}^{n} h_{jk}(t) * i_k(t) = \sum_{k=1}^{n} \int_{-\infty}^{+\infty} h_{jk}(r) i_k(t-r) dr, \qquad j = 1, \ldots, m \tag{5.1}$$

where $h_{jk}(t)$ are the impulse response functions, and "$*$" denotes the convolution operation.

Let $I_j \triangleq \lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} i_j(t) dt$ be the average current in the jth bus branch. Combining this with (5.1) and exchanging the order of the integrals, we get :

$$I_j = \sum_{k=1}^{n} h_{jk}(t) * I_k, \qquad j = 1, \ldots, m \tag{5.2}$$

where we have made use of the fact that $\lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} i_k(t - r_1) dt$ is equal to $I_k \triangleq \lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} i_k(t) dt$ for any given r_1. The proof of this is identical to that of equation (3.3) and assumes the existence of an arbitrary, but finite, upper bound on $i_k(t)$.

In other words, if the time-averages of the contact currents are themselves applied at the contacts, and the bus is solved (i.e., simulated) as a resistive network (DC solution), the resulting branch currents *are* the required time-averages of the bus currents. To complete the solution, we will now express the time-average contact currents I_k in terms of the transition densities inside the circuit.

Let $D(x)$ be the transition density at the output node x of a CMOS gate that draws power supply current $i(t)$ whose time-average is I. Furthermore, let C_n (C_p) be the total capacitance from x to the ground (power) bus connection. These capacitances are the sum of (i) any lumped capacitance tied to the gate output, (ii) MOSFET drain and source capacitances in the gate output stage, and (iii) MOSFET gate capacitances in any logic gates driven by x. As such, they are related to both load capacitance and transistor strength. It has been established [13] that a good estimate of the supply current $i(t)$ can be obtained by looking only at its capacitive charging/discharging component. Since the charge drawn from the supply whenever the gate switches low-to-high (high-to-low) is $V_{dd}C_n$ ($V_{dd}C_p$), it follows that :

$$I \triangleq \lim_{T\to\infty} \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{+T}{2}} i(t)dt = \lim_{T\to\infty} \frac{\frac{1}{2}n_x(T)V_{dd}C_n + \frac{1}{2}n_x(T)V_{dd}C_p}{T} = \frac{V_{dd}C}{2}D(x) \qquad (5.3)$$

where $C \triangleq C_n + C_p$ is the total capacitance at the output node.

Equations (5.3) and (5.2) provide an efficient technique for computing the average current in every branch of the bus, given the transition densities at all circuit nodes. It is significant that this requires only a single DC simulation of the resistive network representing the power bus; no transient simulation is required, and the bus capacitance is irrelevant.

5.2. Average power dissipation

As a direct consequence of the above results, it should be clear that the overall average power dissipation is given by $P_{av} = \frac{1}{2}V_{dd}^2 \sum C_i D(x_i)$, summing over all circuit nodes x_i.

5.3. Electromigration failures

Electromigration [14, 15] is a major reliability problem caused by the transport of atoms in a metal line due to the electron flow. Under persistent current stress, this can cause deformations of the metal, leading to either short or open circuits. The time-to-failure is a lognormally-distributed random variable. It is usually characterized by the *median (*or *mean) time-to-failure* (MTF) [15], which depends on the current density in the metal line.

The models for MTF prediction under pulsed-DC or AC current stress are still controversial. Some recent models [16] predict that, at least under pulsed-DC conditions, the average current is sufficient to predict the MTF, as follows :

$$MTF = \frac{A}{I^2} \qquad (5.4)$$

where A is a parameter that does not depend on the current and I is the average current. However, other recent studies [17] show that the situation is much more complicated.

In any case, even if I is not sufficient by itself to estimate the MTF, it represents a *first order* approximation of the current stress in the wire. Thus equations (5.2) and (5.3), based on the transition density, provide the required average current values I, and help identify potential electromigration problems in the power/ground bus branches.

5.4. Hot electron degradation

As MOSFET devices are scaled down to very small dimensions, certain physical mechanisms start to cause degradation in the device parameters, causing major reliability problems. One such mechanism is the injection of "hot electrons" (or in general hot carriers) into the MOS gate oxide layer [14]. Trapping of these carriers in the gate insulator layer causes degradation in the transistor transconductance and/or threshold voltage.

It is widely accepted that the MOSFET substrate current is a good indicator of the severity of the degradation. In fact one can write an expression for the "age" of a transistor (i.e. how far it is down the degradation path) that has been operating for time T as follows [18] :

$$\text{Age}(T) = \int_{\frac{-T}{2}}^{\frac{+T}{2}} \frac{I_{ds}}{WH} \left[\frac{I_{sub}}{I_{ds}}\right]^m dt \tag{5.5}$$

where $I_{ds}(t)$ & $I_{sub}(t)$ are the MOSFET drain-to-source & substrate currents, W is the channel width, and H and m are parameters that do not depend on the transistor currents.

In order to see how this can be used in a CMOS circuit, consider a MOSFET in a CMOS inverter whose output node is x. It can be shown that the both $I_{sub}(t)$ and $I_{ds}(t)$ are non-zero only when the inverter is switching (this also holds for any CMOS gate). Whenever the inverter switches, it generates two current pulses $I_{sub}(t)$ and $I_{ds}(t)$. The pulses resulting from different switching events are identical except for a dependence on the rise/fall at the inverter input. If one assumes a certain nominal rise/fall time at the input, then using (5.5) one can compute the incremental aging due to $0 \to 1$ and $1 \to 0$ transitions at the inverter output, call these A_{lh} and A_{hl}. Then (5.5) may be written :

$$\text{Age}(T) = (A_{lh} + A_{hl})\,\frac{n_x(T)}{2} \tag{5.6}$$

Degradation due to hot-carriers takes years to manifest itself. In other words, T and $n_x(T)$ are very large, which (using (2.2)) permits the approximation $n_x(T) \approx TD(x)$, and leads to :

$$\text{Age}(T) = \left[\frac{A_{lh} + A_{hl}}{2}\right] TD(x) \tag{5.7}$$

Thus, if CMOS gates are pre-characterized to estimate the incremental damage to their transistors due to a single output transition, then the transition density provides the means to predict transistor aging over extended time periods using (5.7).

6. Experimental Results

We have implemented this approach in a prototype *density simulator*, called DENSIM, that takes a description of a circuit in terms of its Boolean modules and gives the transition density at every node. It also accepts values for transition density and equilibrium probability at the primary inputs. Our current implementation is restricted to combinational (non-feedback) circuits. Every Boolean module should be an instance of a *model* from a simulation library built by a separate *model compiler* called MODCOM. MODCOM uses an input specification in the form of Boolean equations to build a BDD representation of the module outputs and the relevant Boolean differences, and stores this in a model file that DENSIM can use.

We present below the results of a number of test cases that were used to investigate the accuracy and efficiency of this technique. In order to assess the accuracy of the results, we have devised a test by which randomly-generated logic waveforms are fed to the circuit primary inputs and propagated into the circuit (by logic simulation based on the BDDs). The logic simulator uses assignable non-zero delays, scaling them based on the fanout load at every module output. The input waveforms must have the same probability and density values given to DENSIM, and are generated as follows. Starting with $P(x)$ and $D(x)$ values, we solve for μ_0 and μ_1 from (2.3a, b). We then use (arbitrarily) an exponentially distributed random number generator to produce sequences of inter-transition times that have the means μ_0 and μ_1 (the theory presented above holds for any distribution of inter-transition times). Starting from arbitrary initial values, the waveforms are built using these sequences. From the logic simulation results, we estimate the average number of transitions per unit time for every circuit node. For a large number of input transitions, this number should converge to the transition density, according to equation (2.2). We also estimate the fraction of time that the signal spends in the high state and check if that converges to the equilibrium probability, in accordance with (2.1).

In the first few test cases to be presented, the modules were chosen to contain all reconvergent fanout. Thus all signals are independent and the results from DENSIM should agree exactly with those from logic simulation. We will then move on to other test cases where signal correlation does become an issue and will study the speed-accuracy trade-off involved.

As a first test case, consider a single logic module with 8 inputs and one output that implements the Boolean function : $Z = ABFD + CFD + ABHD + CHD + ABFG + CFG + ABHG + CHG + AFE + ADE + CFE + CDE$. Using input values of $P = 0.5$ and $D = 2.0$, DENSIM gives $P(Z) = 0.476562$ and $D(Z) = 3.71875$. The results of the logic simulation run, showing the correct convergent behavior at the output Z, are shown in Fig. 3.

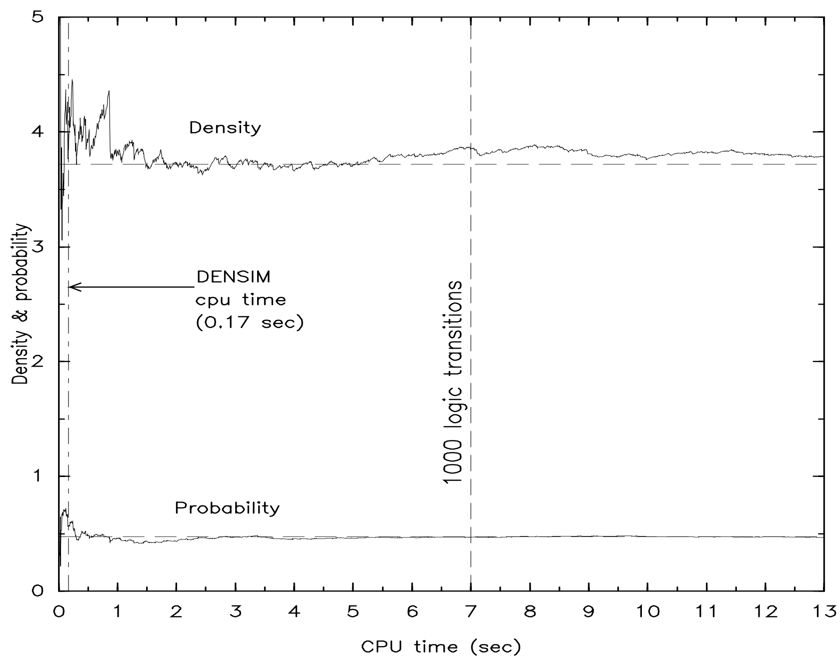

Figure 3. Density and probability convergence plot at node Z.

The horizontal axis in this figure is the cpu time elapsed during the logic simulation run, and the vertical axis is the cumulative values of density and probability at the output node. The two horizontal dashed lines are the values of density and probability computed by DENSIM and the vertical dashed line indicated by the arrow shows the total cpu time required by the DENSIM run. The other vertical line indicates the cpu time required to observe 1000 logic transitions at node Z.

The second test case is the 4-bit ALU/function generator SN54181 from the TI TTL data book. This circuit has 75 logic gates and is shown in Fig. 4.

If we consider the whole circuit as a single Boolean module, then the effects of all internal node correlations are taken care of, and the DENSIM results should, again, be exact. It takes

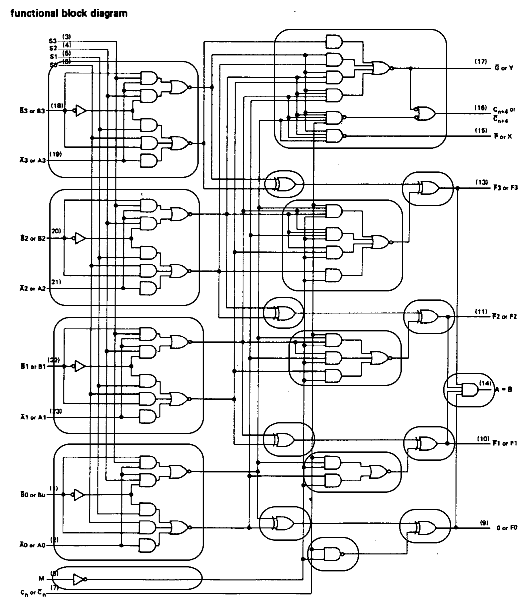

Figure 4. ALU / function generator circuit.

MODCOM 6.53 cpu seconds (SUN SparcStation 1) to build and store the 6092-node BDD model in this case, and DENSIM requires 0.84 cpu seconds (SUN) to run on it. The DENSIM results for the two output nodes $F3$ and X are shown in Figs. 5 and 6, respectively.

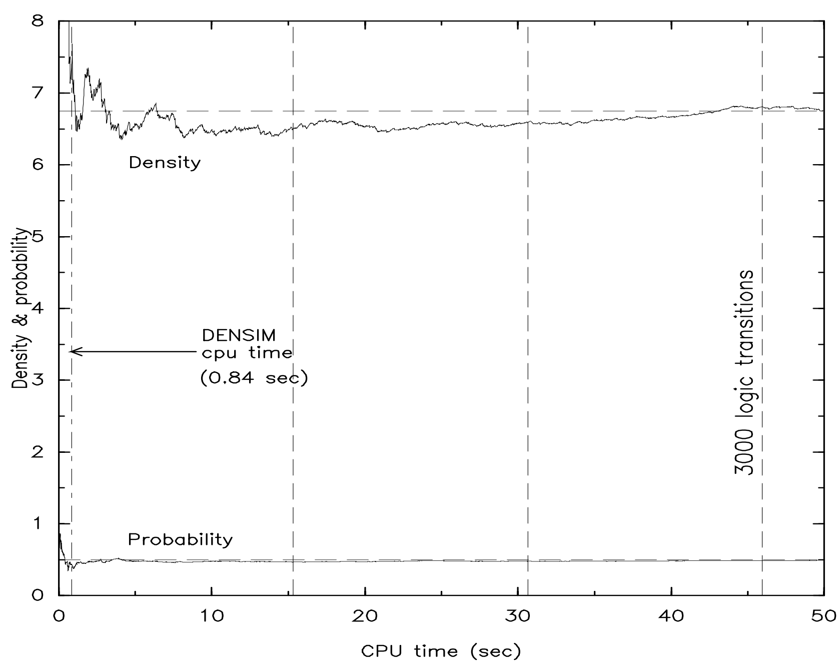

Figure 5. Results for node $F3$ of the ALU.

The preceding test cases show that, even for single-module circuits, computing the den-

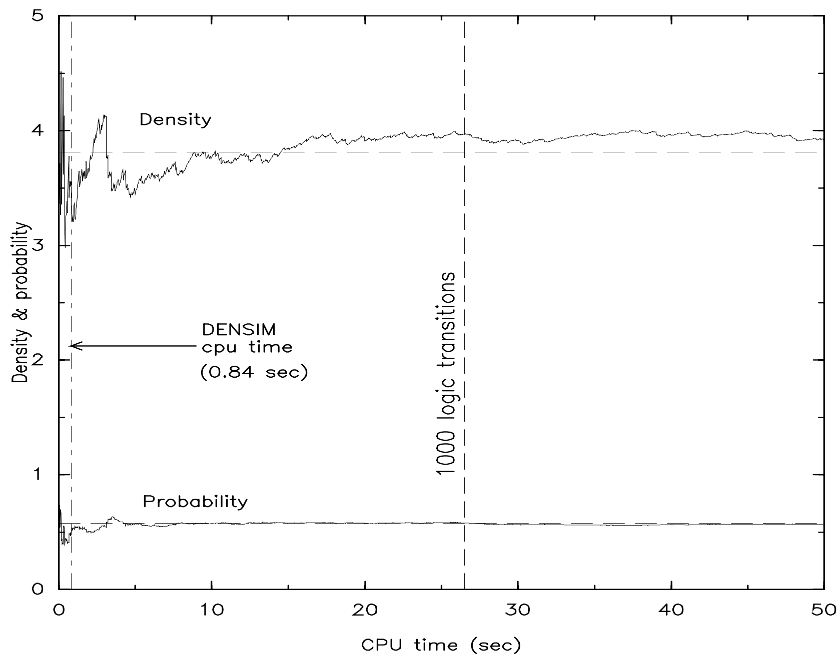

Figure 6. Results for node X of the ALU.

sity values using DENSIM instead of traditional logic simulation is accurate, much faster, and avoids lengthy simulations involving thousands of logic transitions. This observation will be further enforced by the results presented below.

Moving on to multi-module circuits, consider a 32-bit binary ripple adder. In this case, we chose the full-adders to be our Boolean modules. This again leads to a situation where all reconvergent fanout and signal correlation is inside the modules, and where DENSIM results should be exact. DENSIM takes only 0.46 cpu seconds (SUN), as opposed to the 5 minutes required for the logic simulation results to converge, as shown in Figs. 7 and 8, respectively.

An interesting feature of the result in Fig. 7 is the prolonged "flat" part of the curve around 1000 transitions. This illustrates the point made in the introduction that it is impossible to tell before hand exactly when a logic simulation run should be terminated. In this case, if one were monitoring the density values from logic simulation with the intention of terminating the run when the density "converged to something," one might terminate the run at 1000 transitions, getting the wrong result.

We now move on to a consideration of the effects of signal correlation caused by reconvergent fanout. As pointed out in sub-section 4.2, one can accurately handle these effects by keeping all reconvergent fanout within the Boolean modules. However, since large BDDs are

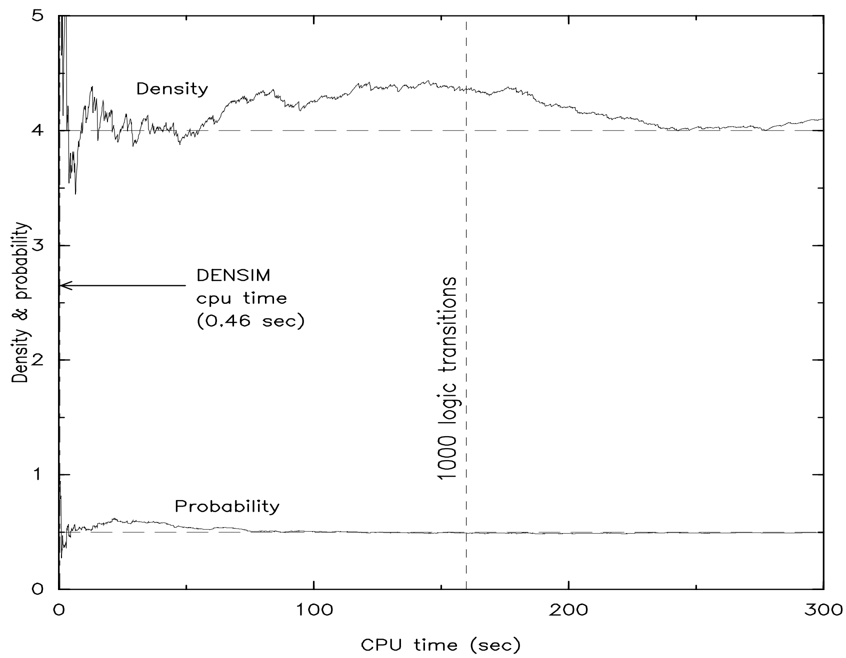

Figure 7. Results for node $n2$ of the adder.

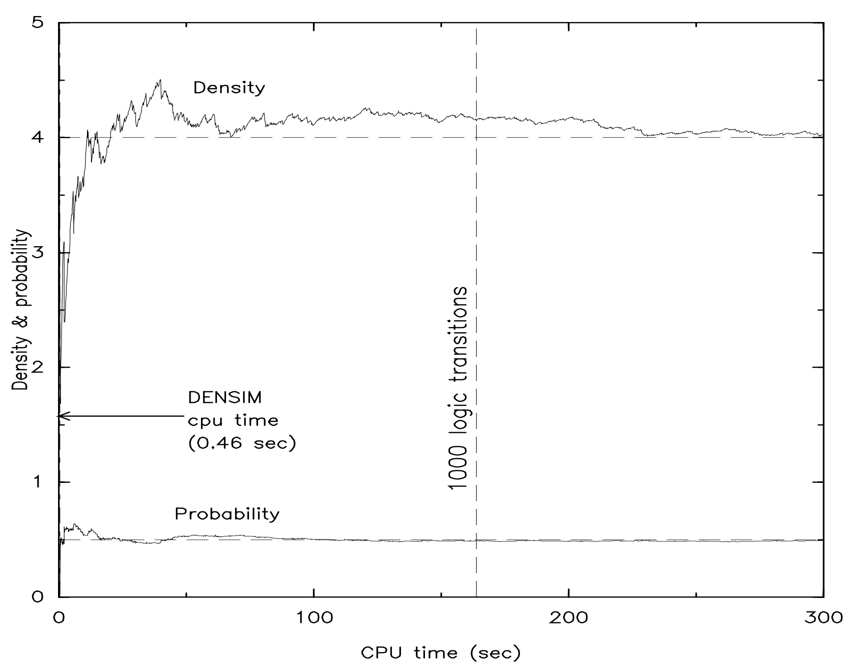

Figure 8. Results for node $n129$ of the adder.

expensive to build and maintain, this can become impractical and leads to a speed-accuracy trade-off. To illustrate this point, we again consider the ALU circuit in Fig. 4. We partition the circuit into the 19 smaller modules shown in the figure and examine the resultant density values at all nodes that are module outputs. By comparing these to the values obtained from the single-module run on this circuit, we get the error histogram shown in Fig. 9. In this case there was a less than 29% loss in accuracy for a 15X gain in speed.

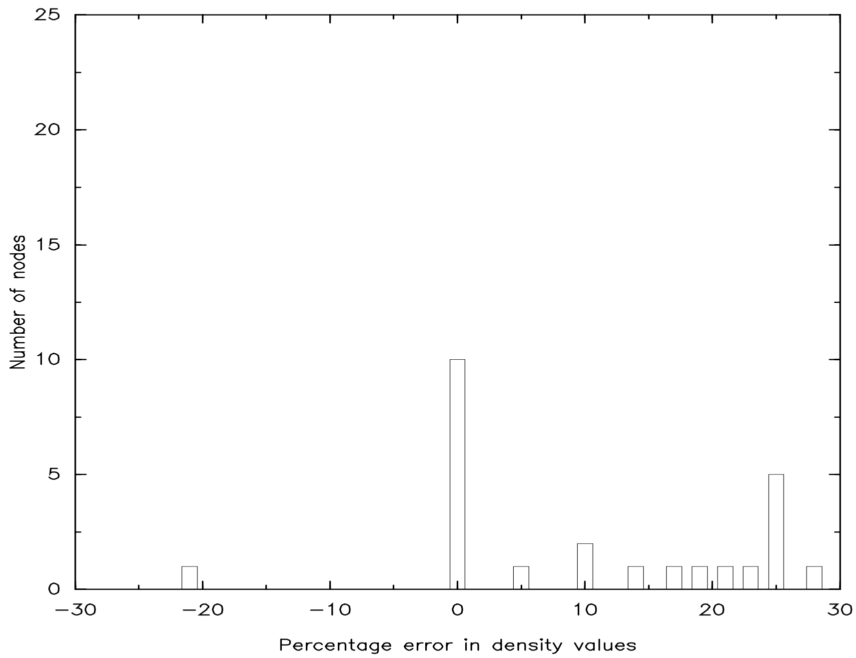

Figure 9. Error histogram for the ALU.

For a further comparison, we ran a logic simulation on the ALU using its gate-level representation, and compared the resulting densities to those observed in the above 19-module run. The error histogram in this case is shown in Fig. 10. All but one of the densities are within 23%. The single point of poor agreement is at node AB which is a reconvergent node for all four ALU outputs F0–F3.

Finally, we present some results obtained for the ISCAS-85 benchmark circuits [19]. In this case we used a "lowest-level partitioning" in which every logic gate was represented as a separate Boolean module. This provides the fastest, but potentially the least accurate, DENSIM run. The 10 ISCAS circuits, their sizes, and the total DENSIM cpu time (on a CONVEX c240) are shown in Table 1.

The execution times are excellent, taking under 10 seconds even for the largest circuit. As for the accuracy, it becomes exceedingly difficult to assess for large circuits, because the BDDs become unacceptably large. Even though BDDs for the these circuits *have* been built by other researchers, the BDDs that we require are much larger because they must

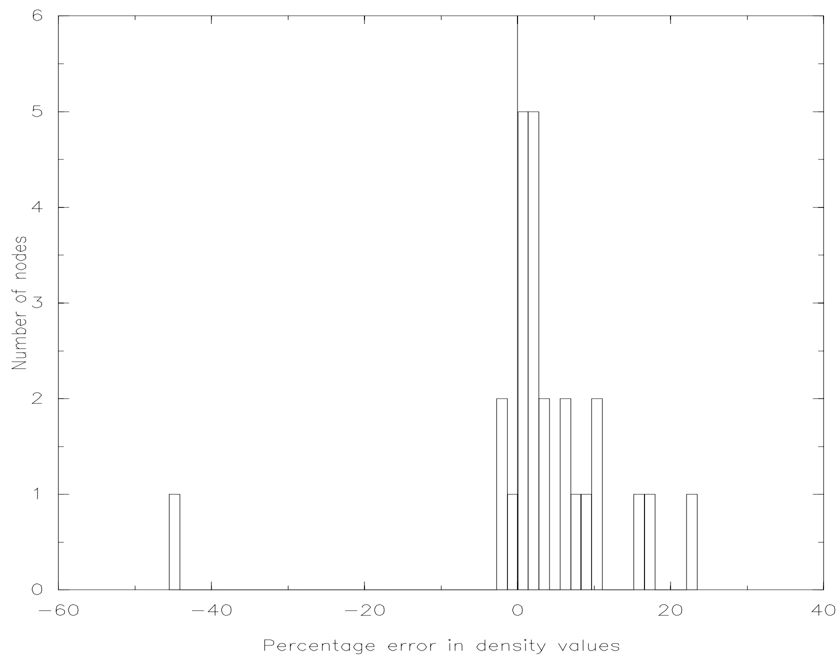

Figure 10. Second error histogram for the ALU.

Table 1. Execution time results for the ISCAS-85 benchmark circuits.

Circuit Name	Size (# gates)	Total Time (cpu sec.)
c432	160	0.52
c499	202	0.58
c880	383	1.06
c1355	546	1.39
c1908	880	2.00
c2670	1193	3.45
c3540	1669	3.77
c5315	2307	6.41
c6288	2406	5.67
c7552	3512	9.85

include the Boolean function at every internal node as well as the output nodes, along with all the associated Boolean difference terms. Thus we are reduced to having to assess the accuracy by obtaining a best-possible estimate of the densities from long logic simulation runs. Even then, it is practically impossible to examine the density plot for every internal node to determine whether the run was long enough for it to converge. Based on several test cases, however, we found that an average of 1000 transitions per input node seems to be enough to approximate most node densities. Such logic simulation runs were performed on all 10 circuits. In order to tabulate the results, we show the *average density* values (averaged over all circuit nodes) in Table 2.

Table 2. Average density results for the ISCAS-85 benchmark circuits. Time is in cpu seconds.

Circuit Name	Avg. Density (DENSIM)	Avg. Density (time) (logic simulation)	%Error
c432	3.46	3.39 (62.8)	+2.1%
c499	11.36	8.75 (241.1)	+29.8%
c880	2.78	3.25 (131.7)	-14.5%
c1355	4.19	6.18 (407.9)	-32.2%
c1908	2.97	5.01 (463.9)	-40.7%
c2670	3.50	4.00 (618.5)	-12.5%
c3540	4.47	4.49 (1082.0)	-0.4%
c5315	3.52	4.79 (1616.0)	-26.5%
c6288	25.10	34.17 (31057.0)	-26.5%
c7552	3.85	5.08 (2713.0)	-24.2%

The third column in the table also lists the total cpu time required (on the CONVEX) to finish the logic simulation run in each case. Even for the smallest circuits, such long simulation runs meant that hundreds of thousands of internal events had to be simulated. Comparing the execution times between tables 1 and 2 clearly demonstrates the speed advantage of this approach (e.g. 5.67 sec. versus 8 hrs. 38 mins. for c6288). As for the average density values, the agreement is very good for c432 & c3540, acceptable for c880 & c2670, and poor for the other circuits. These results highlight the need to better account for signal correlation if one is to obtain consistently good results in the general case.

In general, the problem of estimating equilibrium probabilities, let alone transition densities, is $\mathcal{NP}$-hard. As a result, no single *efficient* solution will work well in *all* cases. The partitioning strategy in general cases, and the speed-accuracy trade-off, are the focus of our continuing research efforts in this area.

7. Summary and Conclusions

To summarize, we have observed that a common thread that runs through most causes of run-time failure is the extent of circuit *activity*, i.e., the rate at which its nodes are switching. We have defined a new measure of circuit activity, called the *transition density*. Based on a stochastic model of logic signals, we have also presented an algorithm to propagate the density from the primary inputs to internal nodes.

To illustrate the practical significance of these results, we have considered four ways in which the density values can be used to study circuit reliability by estimating (1) the average power & ground currents, (2) the average power dissipation, (3) the susceptibility to electromigration failures, and (4) the extent of hot-electron degradation. We have also presented experimental results that demonstrate the practical significance and power of this approach. We envision that the computation of density values inside the circuit can be used as a pre-processing step, and the resulting information applied to these and possibly other reliability problems.

Acknowledgement

I would like to acknowledge the support of Dr. J-H. Chern and Dr. P. Yang at Texas Instruments. I am also thankful to Dr. P. K. Mozumder, also at Texas Instruments, for many fruitful technical discussions. Finally, thanks are due to Karl Brace of Carnegie-Mellon university for providing the BDD package, and to the Microelectronics Center of North Carolina (MCNC) for supplying the ISCAS-85 benchmark circuits.

Appendix A
Existence of $P(x)$ and $D(x)$

A.1. Existence of $P(x)$

Recall the definition (2.1) of the equilibrium probability : $P(x) \triangleq \lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t)dt$. For convenience, we also repeat the statement of proposition 1 :

Proposition 1. *For a logic signal $x(t)$, the limit in (2.1) always exists.*

Proof : Let $\overline{x}_T \triangleq \frac{1}{T}\int_0^T x(t)dt$ be the time-average of $x(t)$ over $[0,T]$; it suffices to show that $\lim_{T\to\infty} \overline{x}_T$ always exists. Notice that $\overline{x}_T \in [0,1]$ and :

$$\frac{d\overline{x}_T}{dT}(T) = \frac{x(T)}{T} - \frac{\overline{x}_T}{T} \tag{A.1}$$

Since both $x(T)$ and $\overline{x}_T$ are bounded, then $\lim_{T\to\infty} \frac{d\overline{x}_T}{dT}(T) = 0$. By the mean value theorem, for any $\Delta > 0$, there exists a $\gamma \in [T, T+\Delta]$ such that $\overline{x}_{T+\Delta} - \overline{x}_T = \frac{d\overline{x}_T}{dT}(\gamma)\Delta$. Therefore :

$$\lim_{T\to\infty} \{\overline{x}_{T+\Delta} - \overline{x}_T\} = \Delta \lim_{\gamma\to\infty} \frac{d\overline{x}_T}{dT}(\gamma) = 0 \tag{A.2}$$

which means that $\lim_{T\to\infty} \overline{x}_T$ exists. ∎

A.2. Existence of $D(x)$

Recall the definition (2.2) of the transition density : $D(x) \triangleq \lim_{T\to\infty} \frac{n_x(T)}{T}$. We also recall a few other definitions : The time between two consecutive transitions of $x(t)$ is called an *inter-transition time*. μ is the average value of all the inter-transition times of $x(t)$. μ_1 (μ_0) is the average of the high (low), i.e., corresponding to $x(t) = 1$ (0), inter-transition times of $x(t)$. It should be clear that $\mu = \frac{1}{2}(\mu_0 + \mu_1)$. In general, there is no guarantee of the existence of μ, μ_0, and μ_1. If the number of transitions in positive time is *finite*, then we say that there is an *infinite* inter-transition time following the last transition, and $\mu = \infty$. A similar convention is made for negative time. μ_f is the average of all the *finite* inter-transition times of $x(t)$. In general, there is also no guarantee of the existence of μ_f. It should be clear, however, that if μ exists, then μ_f also exists and $\mu_f = \mu$. We are now ready to prove proposition 2, which we restate for convenience :

Proposition 2. *Two parts :*

(i) If μ_f exists and is non-zero, then $D(x)$ exists.

(ii) If μ_0 & μ_1 exist, and $\mu \neq 0$, then $D(x)$ exists and we have :

$$P(x) = \frac{\mu_1}{\mu_0 + \mu_1} \qquad \text{and} \qquad D(x) = \frac{2}{\mu_0 + \mu_1} \qquad (A.3a, b)$$

Proof : (i) Suppose that $\mu_f \neq 0$ exists. We first dispose of the special case when $x(t)$ has a finite number of transitions. In that case, $\lim_{T\to\infty} n_x(T)$ is a finite integer value, and $D(x) = 0$.

Another special case is when $x(t)$ has an infinite number of transitions in only one time direction. Without loss of generality, consider that $x(t) = 0$ for all $t < t_0$. If we build another signal $x'(t)$ so that $x'(t) = x(t)$, for $t > t_0$, and $x'(t) = x(t_0 + (t_0 - t))$, for $t < t_0$, then $x'(t)$ has an infinity of transitions in both time directions and it can be shown that $D(x) = \frac{1}{2}D(x')$. Thus, the existence of $D(x)$ is covered by the general case of a signal with an infinity of transitions in both time directions, to be considered next.

In the general case of an infinity of transitions in both time directions, $x(t)$ cannot have an infinite inter-transition time, so that $\mu_f = \mu$. It will simplify the discussion below to refer to μ rather than μ_f. Consider Fig. A1 where, for every T, t_1 is the time of the last transition of $x(t)$ before $-T/2$, t_2 is that of the first transition after $-T/2$, t_3 is that of the last transition before $+T/2$, and t_4 is that of the first transition after $+T/2$.

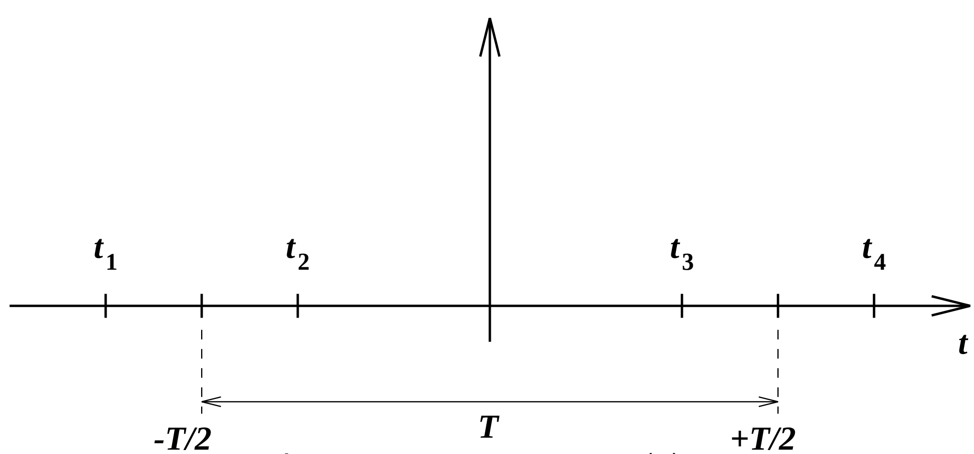

Figure A1. Existence of D(x).

There are $n_x(T)$ transitions between $-T/2$ and $+T/2$, including t_2 and t_3. Thus there are $(n_x(T) - 1)$ inter-transition time intervals between t_2 and t_3. Since $\lim_{T\to\infty} n_x(T) = \infty$, we have :

$$\mu = \lim_{T\to\infty} \frac{t_3 - t_2}{n_x(T) - 1} = \lim_{T\to\infty} \left\{ \frac{n_x(T)}{n_x(T) - 1} \right\} \frac{t_3 - t_2}{n_x(T)} = \lim_{T\to\infty} \frac{t_3 - t_2}{n_x(T)} \qquad (A.4)$$

Likewise :

$$\mu = \lim_{T\to\infty} \frac{t_4 - t_1}{n_x(T) + 1} = \lim_{T\to\infty} \frac{t_4 - t_1}{n_x(T)} \qquad (A.5)$$

We now observe that $t_3 - t_2 \le T \le t_4 - t_1$, which gives :

$$\lim_{T\to\infty} \frac{t_3 - t_2}{n_x(T)} \le \lim_{T\to\infty} \frac{T}{n_x(T)} \le \lim_{T\to\infty} \frac{t_4 - t_1}{n_x(T)} \qquad (A.6)$$

Using (A.4) and (A.5), we see that $\lim_{T\to\infty} \frac{T}{n_x(T)} = \mu$ exists. Since $\mu = \mu_f \neq 0$, then $D(x) = 1/\mu$ exists.

(ii) If μ_0 and μ_1 exist, and $\mu = (\mu_0 + \mu_1)/2$ is non-zero, then μ_f exists and is non-zero and (2.2) exists. Existence of μ_0 and μ_1 also means that $x(t)$ has no infinite inter-transition times, so that $D(x) = 1/\mu$, and we directly get (A.3b) :

$$D(x) = \frac{2}{\mu_0 + \mu_1} \qquad (A.7)$$

To obtain (A.3a), let $n_1(T)$ be the number of (whole) 1-pulses of $x(t)$ in $(\frac{-T}{2}, \frac{+T}{2}]$. It is easy to verify that $\left| n_1(T) - \frac{n_x(T)}{2} \right| \le 1$, which gives $\lim_{T\to\infty} \frac{n_1(T)}{T} = \frac{1}{2} D(x)$. Consider Fig. A2 where, for every T, t_1 is the time of the last $0 \to 1$ transition of $x(t)$ before $-T/2$, t_2 is that of the first $0 \to 1$ transition after $-T/2$, t_3 is that of the last $1 \to 0$ transition before $+T/2$, and t_4 is that of the first $1 \to 0$ transition after $+T/2$.

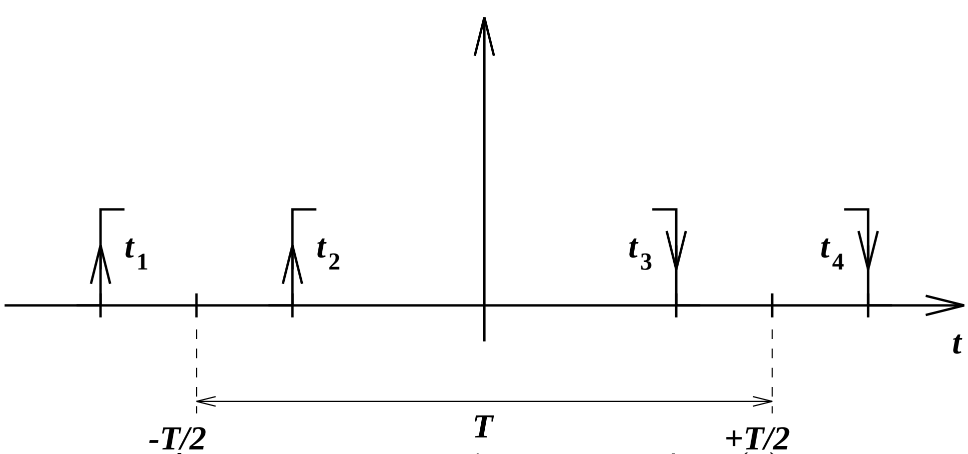

Figure A2. Deriving the expression for $P(x)$.

By definition of μ_1, we have :

$$\lim_{T\to\infty} \frac{1}{n_1(T)} \int_{t_2}^{t_3} x(t)dt = \mu_1 \qquad (A.8)$$

and :

$$\mu_1 = \lim_{T\to\infty} \frac{1}{n_1(T)+2} \int_{t_1}^{t_4} x(t)dt = \lim_{T\to\infty} \left\{ \frac{n_1(T)}{n_1(T)+2} \right\} \frac{1}{n_1(T)} \int_{t_1}^{t_4} x(t)dt \qquad (A.9)$$

which gives :

$$\lim_{T\to\infty} \frac{1}{n_1(T)} \int_{t_1}^{t_4} x(t)dt = \mu_1 \qquad (A.10)$$

We now observe that $\int_{t_2}^{t_3} x(t)dt \le \int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t)dt \le \int_{t_1}^{t_4} x(t)dt$, which gives :

$$\lim_{T\to\infty} \frac{1}{n_1(T)} \int_{t_2}^{t_3} x(t)dt \le \lim_{T\to\infty} \frac{1}{n_1(T)} \int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t)dt \le \lim_{T\to\infty} \frac{1}{n_1(T)} \int_{t_1}^{t_4} x(t)dt \qquad (A.11)$$

Using (A.8) and (A.10), we see that $\lim_{T\to\infty} \frac{1}{n_1(T)} \int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t)dt = \mu_1$. Since $\lim_{T\to\infty} \frac{n_1(T)}{T} = \frac{1}{2}D(x)$, we find that :

$$\mu_1 = \lim_{T\to\infty} \frac{T}{n_1(T)} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} x(t)dt = \frac{2}{D(x)} P(x) \qquad (A.12)$$

which leads to (A.3a) :

$$P(x) = \frac{\mu_1}{\mu_0 + \mu_1} \qquad (A.13)$$

and the proof is complete. ∎

In order to illustrate how mild the condition of proposition 2 is, one can prove another (more stringent) sufficient condition for the existence of $D(x)$, namely, that there exist a non-zero *lower-bound* $\delta_x > 0$ on the inter-transition times. The proof is as follows : Consider the logic signal $x_\delta(t)$ built as follows : $x_\delta(t)$ is 0 everywhere, except on intervals of width δ_x centered at every transition time-point of $x(t)$, where it is 1. It is clear that $\left| n_x(T) - \frac{1}{\delta_x} \int_{\frac{-T}{2}}^{\frac{+T}{2}} x_\delta(t)dt \right| < 1$. Therefore :

$$\lim_{T\to\infty} \frac{n_x(T)}{T} = \frac{1}{\delta_x} \lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{+T}{2}} x_\delta(t)dt \qquad (A.14)$$

By proposition 1, and since $\delta_x > 0$, the density exists. This condition can be easily satisfied in all practical cases.

Appendix B
Using BDDs for Probability Propagation

We will briefly review the concept of a Binary Decision Diagram [10, 11] (BDD) and then present a new application for BDDs as tools for computing the probability of a Boolean function.

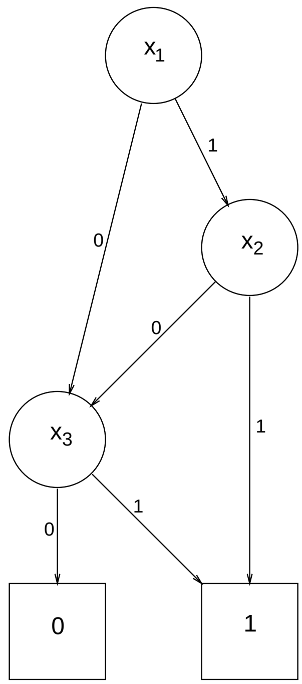

Figure B1. Example BDD representation.

Consider the Boolean function $y = x_1 \cdot x_2 + x_3$, which can be represented by the BDD shown in Fig. B1. The Boolean variables x_i are *ordered*, and each *level* in the BDD corresponds to a single variable. Each level may contain one or more BDD nodes at which one can branch in one of two directions, depending on the value of the relevant variable. For example, suppose that $x_1 = 1$, $x_2 = 0$, and $x_3 = 1$. To evaluate y, we start at the top node, branch to the right since $x_1 = 1$, then branch to the left since $x_2 = 0$, and finally branch to the right since $x_3 = 1$ to reach the terminal node "1". Thus the corresponding value of y is 1.

The importance of the BDD representation is that is is *canonical*, i.e., that it does not depend on the Boolean expression used to express the function. In our case, if the function was expressed as $y = x_3 + x_1 \cdot (x_2 + x_3)$ (an equivalent representation), it would have the same BDD. BDDs have been found to be an efficient representation for manipulating Boolean functions, both in terms of memory and execution time. For example, checking if a Boolean function is satisfiable can be done in time that is linear in the number of variables.

Let $y = f(x_1, \ldots, x_n)$ be a Boolean function. We will show that, given signal probabilities for the variables x_i, and that these variables are independent (random variables), then the probability of the function f can be obtained in *linear time* (in the size of its BDD representation). By Shannon's expansion :

$$y = x_1 f_{x_1} + \overline{x_1} f_{\overline{x_1}} \tag{B.1}$$

where $f_{x_1} = f(1, x_2, \ldots, x_n)$ and $f_{\overline{x_1}} = f(0, x_2, \ldots, x_n)$ are the *cofactors* of f with respect to x_1. Since $x_1 \overline{x_1} = 0$, then :

$$P(y) = P(x_1 f_{x_1}) + P(\overline{x_1} f_{\overline{x_1}}) \tag{B.2}$$

Since the cofactors of x_i do not depend on x_i, and since all variables are independent, then :

$$P(y) = P(x_1) P(f_{x_1}) + P(\overline{x_1}) P(f_{\overline{x_1}}) \tag{B.3}$$

This equation shows how the BDD is to be used to evaluate $P(y)$. The two nodes that are descendants of y in the BDD correspond to the cofactors of f. The probability of the cofactors can then be expressed in the same way, in terms of their descendants. Thus a depth-first-traversal of the BDD, with a post-order evaluation of $P(\cdot)$ at every node is all that is required. We have implemented this using the "scan" function of the BDD package [11].

References

[1] F. Najm, "Transition density, a stochastic measure of activity in digital circuits," *28th ACM/IEEE Design Automation Conference*, pp. 644–649, June 17–21, 1991.

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

[3] E. Parzen, *Stochastic Processes*, San Francisco, CA: Holden-Day Inc., 1962.

[4] D. R. Cox & H. D. Miller, *The Theory of Stochastic Processes*, New York: John Wiley & Sons Inc., 1968.

[5] K. P. Parker and E. J. McCluskey, "Probabilistic treatment of general combinational networks," *IEEE Transactions on Computers*, pp. 668-670, June 1975.

[6] S. C. Seth, L. Pan, and V. D. Agrawal, "PREDICT - probabilistic estimation of digital circuit testability," *IEEE 15th Annual International Symposium on Fault-Tolerant Computing*, pp. 220-225, June 1985.

[7] J. Savir, G. S. Ditlow, and P. H. Bardell, "Random pattern testability," *IEEE Transactions on Computers*, pp. 79–90, January 1984.

[8] G. Markowsky, "Bounding signal probabilities in combinational circuits," *IEEE Transactions on Computers*, pp. 1247–1251, October 1987.

[9] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Riccó, "Estimate of signal probability in combinational logic networks," *1989 IEEE European Test Conference*, pp. 132–138, 1989.

[10] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computer-Aided Design*, pp. 677–691, August 1986.

[11] K. S. Brace, R. L. Rudell, and R. E. Bryant, "Efficient implementation of a BDD package," *27th ACM/IEEE Design Automation Conference*, pp. 40–45, June 1990.

[12] F. N. Najm, R. Burch, P. Yang, and I. N. Hajj, "Probabilistic simulation for reliability analysis of CMOS VLSI circuits," *IEEE Transactions on Computer-Aided Design*, pp. 439–450, April 1990.

[13] H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE Journal of Solid-State Circuits*, vol. SC-19, no. 4, pp. 468–473, Aug. 1984.

[14] M. H. Woods, "MOS VLSI reliability and yield trends," *Proc. of the IEEE*, vol. 74, no. 12, Dec. 1986.

[15] J. R. Black, "Electromigration failure modes in aluminum metallization for semiconductor devices," *Proc. of the IEEE*, vol. 57, no. 9, Sep. 1969.

[16] B. K. Liew, N. W. Cheung, and C. Hu, "Electromigration interconnect lifetime under AC and pulsed DC stress," *IEEE International Reliability Physics Symposium*, pp. 215–219, April 1989.

[17] J. S. Suehle and H. A. Schafft, "Current density dependence of electromigration t_{50} enhancement due to pulsed operation," *IEEE International Reliability Physics Symposium*, pp. 106–110, April 1990.

[18] C. Hu, "Reliability issues of MOS and bipolar ICs," *IEEE International Conference on Computer Design*, pp. 438–442, 1989.

[19] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran," *IEEE International Symposium on Circuits and Systems*, pp. 695–698, June 1985.