Estimation of Circuit Activity Considering Signal Correlations and Simultaneous Switching

Tan-Li Chou
Electrical Engineering
Purdue University
West Lafayette, IN

Kaushik Roy
Electrical Engineering
Purdue University
West Lafayette, IN.

Sharat Prasad
Integrated Systems Lab.
Texas Instruments, Inc.
Dallas, TX.

Abstract
This paper presents accurate estimation of signal activity at the internal nodes of CMOS combinational logic circuits. The methodology is based on a stochastic model of logic signals and takes into account signal correlations and simultaneous switching of signals at logic gate inputs into consideration. In combinational logic synthesis, in order to minimize spurious transitions due to finite propagation delays, it is crucial to balance all signal paths and to reduce the logic depth [4]. As a result of balancing delays through different paths, the inputs to logic gates may switch at approximately the same time. We have developed and implemented an analytical technique to calculate signal probability and switching activity of the CMOS combinational logic circuits. Experimental results show that if simultaneous switching is not considered the switching activities of the internal nodes can be off by more than 100% compared to simulation based techniques. In contrast, our technique is on the average within 2% of logic simulation results.

1 Introduction
With the recent trend toward portable computing and communication systems there has been an increasing thrust toward considering power dissipation during VLSI design [4, 3, 8]. In order to design circuits for low power and high reliability, accurate estimation of power dissipation is required. In CMOS circuits, the majority of the power dissipation is due to charging and discharging of load capacitance of logic gates. Such charging and discharging occurs due to signal transitions. The problem of determining when and how often transitions occur at a node in a digital circuit is difficult because they depend on the applied input vectors and the sequence in which they are applied. Therefore, probabilistic techniques have been resorted to. Research directed at estimating signal activity for combinational logic, as reported in [2, 5, 6]. However, such methods fail to consider the effect of "near simultaneous" signal switching at logic gate inputs. SPICE simulation results shown in Figure 2 for the circuit of Figure 1 shows that the spurious pulses generated at node V1 is negligible at node V4 if the two primary inputs have a rising and a falling transition respectively, within 3ns of each other. The spurious pulses try to charge or discharge the capacitances associated with the nodes of a circuit. If such pulses are not wide enough to charge or discharge the capacitances, they disappear. The above example shows that if the inputs to a logic gate switch within a period of \( \Delta t \), spurious transitions do not occur at the output. \( \Delta t \) is a function of the inertial delay of the gate and the load capacitances associated with the gate.

"ONE"

![Figure 1: A Circuit for SPICE simulation](image1)

![Figure 2: Timing diagram for SPICE simulation](image2)

![Figure 3: Example of simultaneous switching](image3)
2 Preliminaries and Definitions

In this section we describe the representation of multi-level circuits and review the concept of signal probability and activity, followed by a brief discussion on power consumption in CMOS logic circuits.

Multi-level Logic Representation: Multi-level logic can be described by a set $F$ of completely specified Boolean functions. Each Boolean function $f \in F$ maps one or more inputs and intermediate signals to a non output or a new intermediate signal. A circuit is represented as a Boolean network. Each node has a Boolean variable and a Boolean expression associated with it. One can view this as a set of gates. Each gate has one or more input pins and (generally) one output pin. Several pins are electrically tied together by a signal. Each signal connects to the output pin of exactly one gate, called the driver gate.

Signal Probability and Activity in CMOS: We briefly describe the model used in [2] for estimation of signal activity of circuits in the following. The primary inputs to a combinational circuit are modeled as mutually independent SSS (Strict-Sense Stationary) mean-ergodic $\delta t$ processes. Under this assumption, the probability of the primary input logic signals $x_i(t), i = 1 \ldots n$, assuming the logic value ONE at any given time $t$ becomes a constant, independent of time, and is called the equilibrium probability of the random signal $x_i(t)$. This is denoted by $P(x_i)$. The activity $A(x_i)$ at a primary input $x_i$ of the module is defined as $\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} x_i(t)$ and equals the expected value $\mathbb{E}[x_i]$ of the signal $x_i$. The variable $n_x$ is the number of switching $x_i(t)$ in the time interval $[-T/2, T/2]$. Since digital circuits can be thought of as non-linear but time-invariant systems, the signals at the internal and output nodes of the circuit are also SSS and mean-ergodic. Further, the Boolean functions describing the outputs of a logic module are decoupled from the delays inside the module by assuming the signals to have passed through a special delay module prior to entering the module under consideration. Therefore, the task of propagating equilibrium probabilities through the module is transformed into that of propagating signal probabilities. Also the activities $A(y_j)$ at the nodes $y_j$ is given by

$$A(y_j) = \sum_{i=1}^{n} P \frac{\partial x_i}{\partial y_j} A(x_i)$$

Here $\frac{\partial x_i}{\partial y_j}$ is the Boolean difference of function $y$ with respect to $x$ and it is defined by $\frac{\partial x_i}{\partial y_j} = y \mid_{x_i=1} - y \mid_{x_i=0}$.

Equation 1 considers signal correlations within a logic module, it does not take simultaneous switching into account. Hence, this method results in errors in estimating activities of a circuit.

In order to better explain our approach, we introduce the concept of normalized activity, which will be used in Section 3. Assume all primary inputs to the module switch only at the leading edge of the clock and the module is delay-free. Every signal $x$ at the internal or output node of the circuit becomes a discrete-time stochastic process. Therefore, $x(t)$ represents the logic value for the time interval $[t, t+T]$, where $t$ is some leading edge of the clock cycle and $T$ is the clock cycle of the circuit module. This is due to the fact that during the interval the signal $x(t)$ does not change value. We denote $P(x(t))$ as the probability of node $x$ being logic ONE in the time interval $[t, t+T]$. If one selects a clock cycle at random, the probability of having a switching at $t$ at node $y$ is $A(y)/f$. Here $A(y)$ is the activity at node $y$ and $f$ is the clock frequency [7]. We define the normalized activity $a(y)$ as $A(y)/f$.

3 Accurate Activities

When more than one primary input, say $x_i$ and $x_j$, are switching simultaneously, the Boolean differences $\frac{\partial x_i}{\partial y_j}$ and $\frac{\partial x_j}{\partial y}$ are undefined at these time instants. Hence, the proof in [2] that leads to equation 1 is no longer valid for this situation. Therefore, we need the following definitions to derive the accurate expression of activity considering simultaneous switching of signals.

Let $y$ be a Boolean expression and $x_i, i = 1 \ldots n$ be mutually independent primary inputs of $y$. We define

$$a_{\delta t}^{(x_1, \ldots, x_n)} = y \mid_{x_1=1} - y \mid_{x_1=0} \cdot \ldots \cdot y \mid_{x_n=1} - y \mid_{x_n=0}$$

where $k$ is positive integer, $i_k$ is logic value ONE or ZERO and $x_i, j = 1 \ldots k$ are distinct mutually independent primary inputs of $y$. We define $P(x_1, \ldots, x_n) = P(x_1|x_2, \ldots, x_n) = \ldots = P(x_k|x_1, \ldots, x_{k-1})$ as the conditional probability of having a transition at time $t$ while $x_1, x_2, \ldots, x_k$ are switching at time $t$ and the rest of the signals are not. $t$ is some leading edge of the clock. Under this condition, the probability of $x_1$ being ONE in the time interval $[t, t+T]$ is $\frac{t}{\delta t}$ rather than $P(x_j)$, where $T$ is the clock cycle.

This is due to the fact that we have assumed $x_j$ is not switching at the same time $t$ and can be explained as follows. $x_j(t)+x_j(t) = 1$ and $x_j(t)+x_j(t) = 1$ signify that $x_j$ does not switch at time $t$ and remain ONE and ZERO respectively. Similarly, $x_j(t)+x_j(t) = 1$ and $x_j(t)+x_j(t) = 1$ signify that $x_j$ has a transition from ONE to ZERO and from ZERO to ONE respectively. Therefore, since $x_j$ is SSS, the following equations hold:

$$P(x_j(t)+x_j(t)|x_j(t)+x_j(t) = 1 = a(x_j))$$

$$P(x_j(t)+x_j(t)|x_j(t)+x_j(t) = 1 = a(x_j))$$

However, because $P(x_j(t)+x_j(t) = 1)$ since $x_j$ is SSS, and $P(x_j(t)+x_j(t) = 1)$, it follows that

$$P(x_j(t)+x_j(t) = 1|x_j(t)+x_j(t) = 1) = \frac{t}{\delta t}$$

Therefore, $P(x_j(t)+x_j(t) = 1|x_j(t)+x_j(t) = 1) = \frac{t}{\delta t}$.

Since $x_j(t)+x_j(t) = 1$ and $x_j(t)+x_j(t) = 1$, it follows that

$$P(x_j(t)+x_j(t) = 1|x_j(t)+x_j(t) = 1) = \frac{t}{\delta t}$$

The conditional probability of $x_j$ being ONE in the time interval $[t, t+T]$ while it does not switch at $t$, denoted as $P(x_j)$, is given by

$$P(x_j(t) = 1|x_j(t) = 1|x_j(t) = 1) = \frac{t}{\delta t}$$

Under the assumption that the primary inputs are mutually independent and the logic signals can be modeled as strictly stationary (SSS) mean-ergodic 0-1 discrete-time stochastic processes with logic modules having zero delays, the following-
Theorem 1 (n-inputs) If \( y \) is a Boolean expression and \( x_1, i = 1 \ldots n \) are mutually independent primary inputs of \( y \). Then
\[
a(y) = \sum_{x_i} P(x_i) \prod_{1 \leq i < j \leq n} (1 - a(x_j))
\]
\[
+ \frac{1}{2} \sum_{1 \leq i < j \leq n} \left[ P(x_i) \cdot a(x_j) \cdot \left( \frac{a(x_i) + a(x_j) - 1}{2} \right) \right] \prod_{1 \leq i \leq n} a(x_i)
\]
\[
+ \prod_{1 \leq i < j \leq n} (1 - a(x_j)) \prod_{1 \leq i \leq n} a(x_i)
\]
where \( P(x_i) \cdot a(x_j) \cdot \left( \frac{a(x_i) + a(x_j) - 1}{2} \right) \) are conditional probabilities under the condition that only some primary inputs switch and the rest do not at the leading edge of the clock cycle.

Hence, if we apply Theorem 1 to calculate the exact activity, we only have to compute
\[
P\left( \frac{a(x_i) + a(x_j) - 1}{2} \right) \prod_{1 \leq i \leq n} a(x_i)
\]
This can be CPU time intensive. In the next section, we will show how to utilize symbolic probability to calculate exact activity.

4 Derivation of Activities from Signal Probabilities

Often times symbolic probability expression in terms of input symbolic probabilities or Binary Decision Diagram of a node is created before activity is computed [2, 3, 5, 8]. Therefore, it saves time and memory space by deriving the activity from its symbolic probability expression. We first show how to compute the exact activity, considering simultaneous switching.

We know from the last section that \( P(y)(t-T) | y(t) = 1 \) is the Boolean expression of the primary inputs excluding \( y \) where \( y \) is some node in a circuit. Hence, \( a(y) = 2P(y) - P(y)(t-T) | y(t) = 1 \).

Instead of computing \( P(y)(t-T) | y(t) = 1 \) directly, we can utilize the symbolic form of \( P(y) \) to do the same. Let \( y = f(x_1, x_2, \ldots, x_n) \) be a Boolean expression in terms of the primary inputs. We can express \( y(t-T) | y(t) = 1 \) as follows,
\[
y(t-T) | y(t) = 1 = \sum_{x_j} P(x_j) \prod_{1 \leq i < j \leq n} f(x_i, x_j, x_j(t-T), x_i | y(t) = 1)
\]
where \( f(.) \) is some Boolean expression. Similar to \( y \), \( P(y)(t-T) | y(t) = 1 \) can be expressed as a sum of primary input probability products \( \sum_{x_j} P(x_j) \prod_{1 \leq i < j \leq n} f(x_i, x_j, x_j(t-T), x_i | y(t) = 1) \).

Thus, to calculate the activity of node \( y \) considering signal correlations, all we need is the set of 4 independent inputs, \( y_1, y_2, y_3, y_4 \) and \( y_5 \) is some integer. Note that \( y_1, y_2, y_3, y_4 \) do not include \( y_5 \).

Therefore, if we define
\[
P(x)(t-T) | y(t) = 1 = P(x)(t-T) f(x)(t) = P(x)(t-T) f(x)(t) = \sum_{x_j} P(x_j) \prod_{1 \leq i < j \leq n} f(x_i, x_j, x_j(t-T), x_i | y(t) = 1),
\]
then we do not need to explicitly compute \( P(y)(t-T) | y(t) = 1 \).

The following example demonstrates this method.

Example 1 \( y = x_1 + x_2 \). Given \( P(y) = P(x_1) + P(x_2) \), where \( P(x_i) = 1 - P(x_i) \) calculate \( P(y)(t-T) | y(t) = 1 \).

\[
P(y)(t-T) | y(t) = 1 = \frac{P(x_1)(t-T) + P(x_2)(t-T) + P(x_1)(t-T) + P(x_2)(t-T)}{2}
\]

Figure 4: the minimum independent inputs to node \( y \)

After rearranging and simplifying terms, we have
\[
a(y) = 1 - \left( 1 \cdot P(x_1) + 1 \cdot P(x_2) \right) \cdot \sum_{x_j} P(x_j(t-T) | y(t) = 1) - 1 \cdot 2 \cdot 2 \cdot 2 \cdot \sum_{x_j} P(x_j(t-T) | y(t) = 1)
\]

Though \( P(y)(t-T) | y(t) = 1 \) is not explicitly expressed in terms of \( P(x_1(t-T)) \) and \( P(x_2(t-T)) \), it needs no extra memory space other than \( P(y) \). This is because they have the same symbolic form.

5 Implementation and Results

The proposed methodology uses signal probability measure to accurately estimate activity for CMOS circuits. It is important to accurately calculate signal probability for further use in estimating activity. We choose the general algorithm proposed in [1] and adopt a data structure similar to [2].

However, the exact calculation of signal probability is \( N P \)-hard [2]. Without properly partitioning the circuits, the method may not be applicable to very large circuits. This is due to the fact that the size of the symbolic probability expression grows exponentially with respect to the number of independent inputs to the circuit module. A partitioning algorithm that limits the number of inputs to a module has been proposed in [9]. However, it does not utilize the information of the circuit topology. For example, let us consider the circuit of Figure 4. Our goal is to calculate the signal probability at node \( y \) considering signal correlations. All we need is the set of 4 independent inputs, \( y_1, y_2, y_3, y_4 \) rather than the whole set of \( y \) primary inputs \( x_1, x_2, \ldots, x_7 \). This is because \( y_1, y_2, y_3, y_4 \) are independent. We, therefore, apply the heuristic similar to [9] to the smaller set and set the maximum number of independent inputs to be 10.

The Probability and Activity Simulator (PAS) to estimate activities at the internal and output nodes of CMOS logic circuit has been implemented in C on the Berkeley SIS environment. We assume that the primary input signal probabilities and activities are available to us through system level simulation of the environment that the circuit will be working in. We first present a number of test cases that show the importance of considering simultaneous switching for circuits. Then we show the accuracy and efficiency of our technique on ISCAS benchmark circuits. In order to assess the accuracy of the results, we use a logic simulator with zero-delay model. We generated 10,000 random primary inputs (conforming to the given probabilities and activities) for logic simulation to determine the activities at the internal nodes of a circuit. The primary input signal probabilities and activities of the circuit were used in PAS to generate \( P(y) \) and \( P(y)(t-T) \) at internal nodes of the circuits.

Table 1 shows the detailed results for an MCNC benchmark example (parity). All inputs were assigned a signal probability of 0.5. All primary inputs were assigned the same activity as...
shown in the table \((act)\). PAS denotes our technique for calculation of signal activity taking simultaneous switching and signal correlations into account. NSS (No Simultaneous Switching) denotes the technique that considers signal correlations but neglects simultaneous switching of signals at logic gate inputs. \(\Phi\) represents the normalized power dissipation measure introduced in Section 2 and is used to compare the results with logic simulation technique \((Sim)\). The percentage error \((Err \%)\) represents the deviation from the simulation results. The CPU times are also shown in the table for a SPARC 10 workstation. It can be observed that the power dissipation measure \(\Phi\) can be off by more than 95\% if simultaneous switching is not considered, while the results for our technique are remarkably close to the simulation results. Figure 5 shows the accuracy of activity calculation for the \(parity\) example. The \(x\)-axis represents the different nodes of the circuit, while the \(y\)-axis represents the normalized activities associated with each node. Node 0 through 15 are the primary inputs. Nodes 16 through 30 are either intermediate nodes or primary outputs. It can be easily observed that the PAS closely follows simulation results, while the errors introduced by NSS can be large.

Table 2 shows the results on 10 ISCAS benchmarks. Results show that the power dissipation measure \(\Phi\) determined by PAS is on the average 2\% of logic simulation results. On the other hand, NSS is at best 21\% off the logic simulation results. It confirms again that simultaneous switching must be considered while estimating power dissipation.

6 Conclusions

By considering signal correlations and “near simultaneous” switching of inputs to static logic gates, we have shown that the activities at internal nodes are remarkably close to the simulation results. But if such switching is not considered, activities can be off by more than 100\%. A technique that derive activity at a static logic gate from symbolic probability has been demonstrated. By partitioning the circuit properly, we have achieved large speed-up in our estimation algorithm while maintaining the accuracy. Hence, this technique can be efficiently used in a logic synthesis environment to estimate power dissipation.

References


