Table 1: The TAG table for a two-input AND gate.

<table>
<thead>
<tr>
<th>OUTPUT TAGS</th>
<th>INPUT TAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,0)</td>
<td>(0,0), (0,0), (0,0), (0,1), (0,0), (1,0), (0,0), (1,1), (0,1), (0,0)</td>
</tr>
<tr>
<td>(0,1)</td>
<td>(0,1), (0,1), (0,1), (1,1), (0,1), (1,0), (1,0), (1,0), (1,1), (0,0)</td>
</tr>
<tr>
<td>(1,1)</td>
<td>(1,1), (1,1), (1,1), (1,1), (1,1), (1,1), (1,1), (1,1), (1,1), (1,1)</td>
</tr>
</tbody>
</table>

Table 2: Comparison of the Power Estimation and Run-time under a unit delay model: ss : using symbolic simulation, tp : using transition probability.

<table>
<thead>
<tr>
<th>circuit</th>
<th>POWER ESTIMATION</th>
<th>RUN TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ss tp % err.</td>
<td>ss tp speed up</td>
</tr>
<tr>
<td>s419</td>
<td>263.3 263.4 0.0</td>
<td>18.5 2.0 9.4</td>
</tr>
<tr>
<td>s512</td>
<td>291.4 294.8 0.1</td>
<td>10.1 1.9 5.6</td>
</tr>
<tr>
<td>s516</td>
<td>324.2 323.3 0.5</td>
<td>11.0 2.4 4.5</td>
</tr>
<tr>
<td>s440</td>
<td>303.2 306.3 0.1</td>
<td>11.1 1.9 5.8</td>
</tr>
<tr>
<td>s444</td>
<td>329.2 326.5 0.8</td>
<td>15.5 2.4 6.4</td>
</tr>
<tr>
<td>s526</td>
<td>449.9 452.9 0.6</td>
<td>16.8 3.1 5.4</td>
</tr>
<tr>
<td>s641</td>
<td>232.0 323.0 0.2</td>
<td>100.2 3.9 23.7</td>
</tr>
<tr>
<td>s713</td>
<td>342.4 341.0 0.4</td>
<td>95.0 4.2 2.9</td>
</tr>
<tr>
<td>s520</td>
<td>464.4 411.3 0.9</td>
<td>25.0 6.0 4.9</td>
</tr>
<tr>
<td>s512</td>
<td>86.9 94.2 0.8</td>
<td>31.9 6.1 5.2</td>
</tr>
<tr>
<td>s553</td>
<td>475.5 473.3 0.4</td>
<td>85.9 6.3 13.5</td>
</tr>
<tr>
<td>s619</td>
<td>83.7 84.3 1.0</td>
<td>283.3 13.4 19.7</td>
</tr>
<tr>
<td>s1238</td>
<td>92.8 92.1 0.8</td>
<td>640.4 12.7 30.4</td>
</tr>
<tr>
<td>s543</td>
<td>1321.5 1322.1 0.0</td>
<td>87.4 11.1 7.8</td>
</tr>
<tr>
<td>s194</td>
<td>1339.4 1337.9 0.1</td>
<td>30.0 11.3 7.0</td>
</tr>
<tr>
<td>s5378</td>
<td>N/A 2167.1  -</td>
<td>N/A 30.3 -</td>
</tr>
<tr>
<td>s838</td>
<td>N/A 476.7  -</td>
<td>N/A 6.8 -</td>
</tr>
<tr>
<td>AVE.</td>
<td>% ERR. 0.66</td>
<td>AVE. 12.9</td>
</tr>
</tbody>
</table>

Table 3: Comparison of the Power Estimation and Run-time under a fixed binary inertial delay model: ss : using symbolic simulation, tp : using transition probability.

<table>
<thead>
<tr>
<th>circuit</th>
<th>FIXED BINARY INERTIAL DELAY</th>
<th>RUN TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>POWER ESTIMATION</td>
<td>RUN TIME</td>
</tr>
<tr>
<td></td>
<td>ss tp % err.</td>
<td>ss tp speed up</td>
</tr>
<tr>
<td>s419</td>
<td>231.0 221.0 0.02</td>
<td>11.1 2.2 2.9</td>
</tr>
<tr>
<td>s512</td>
<td>215.3 210.9 1.04</td>
<td>6.1 2.0 3.2</td>
</tr>
<tr>
<td>s516</td>
<td>311.1 309.3 0.60</td>
<td>6.0 2.5 2.4</td>
</tr>
<tr>
<td>s440</td>
<td>286.6 285.6 1.00</td>
<td>6.6 2.1 3.1</td>
</tr>
<tr>
<td>s444</td>
<td>312.1 308.7 1.26</td>
<td>8.7 2.1 3.2</td>
</tr>
<tr>
<td>s526</td>
<td>426.3 427.8 0.35</td>
<td>7.5 3.4 2.2</td>
</tr>
<tr>
<td>s641</td>
<td>299.1 285.0 3.69</td>
<td>31.5 4.6 6.9</td>
</tr>
<tr>
<td>s113</td>
<td>314.3 304.2 3.25</td>
<td>35.9 3.1 1.0</td>
</tr>
<tr>
<td>s520</td>
<td>201.0 199.8 0.52</td>
<td>17.8 9.9 2.6</td>
</tr>
<tr>
<td>s553</td>
<td>81.0 81.3 0.25</td>
<td>20.0 6.6 3.8</td>
</tr>
<tr>
<td>s593</td>
<td>434.5 426.3 1.43</td>
<td>33.8 7.0 4.2</td>
</tr>
<tr>
<td>s1196</td>
<td>628.2 624.2 1.13</td>
<td>110.4 15.9 7.1</td>
</tr>
<tr>
<td>s1238</td>
<td>832.2 828.3 1.36</td>
<td>145.4 14.7 9.0</td>
</tr>
<tr>
<td>s1488</td>
<td>202.5 201.8 0.06</td>
<td>44.3 12.3 3.6</td>
</tr>
<tr>
<td>s1491</td>
<td>218.6 215.2 0.28</td>
<td>418.5 12.6 33.3</td>
</tr>
<tr>
<td>s5378</td>
<td>2198.0 223.5 2.98</td>
<td>298.4 33.3 8.6</td>
</tr>
<tr>
<td>s838</td>
<td>N/A 494.6  -</td>
<td>N/A 7.6 -</td>
</tr>
<tr>
<td>AVE.</td>
<td>% ERR. 1.39</td>
<td>AVE. 6.0</td>
</tr>
</tbody>
</table>

significant reduction in run time is achieved while maintaining a high degree of accuracy. In the future, we will extend this work to directly handle complex gate, consider non-equal rise and fall times, and estimate power consumption in sequential circuits.

References
where

\begin{align*}
sp_n^{(x,y)}(t^-) &= \sum_{w_i \in W_n^{(x,y)}} P_n(u_i) s v_{w_i}(t^-), \\
sp_n^{(x,y)}(t^+) &= \sum_{w_i \in W_n^{(x,y)}} P_n(u_i) s v_{w_i}(t^+), \\

\end{align*}

and \( S = \{(0,0),(0,1),(1,0),(1,1)\} \). Similarly, (4) and (5) become:

\begin{align*}
t_a(t) &= \sum_{(z,y) \in S} t_u^{(z,y)}(t), \\
t_d(t) &= \sum_{(z,y) \in S} t_d^{(z,y)}(t),
\end{align*}

where

\begin{align*}
t_u^{(z,y)}(t) &= \sum_{w_i \in W_n^{(z,y)}} P_n(u_i) s p_{w_i}(t), \\
t_d^{(z,y)}(t) &= \sum_{w_i \in W_n^{(z,y)}} P_n(u_i) d o_{w_i}(t).
\end{align*}

We refer to \( t_u^{(z,y)}(t) \) and \( t_d^{(z,y)}(t) \) as the \((x,y)\)-tagged transition probabilities of \( n \) at time \( t \).

The tag of a waveform at the output of a two-input AND gate depends on the tags of waveforms at its inputs. Table 1 gives the input tag pair that will force the specified output tag for a two-input AND gate. For example, a waveform at the output with tag \((1,1)\) requires both input waveforms to have \((1,1)\) tags. The set of input tag pairs that forces the output tag to be \( \sigma \) is denoted as the forcing set of \( \sigma \) (FS(\( \sigma \))).

\section*{3.3 Power Estimation using Transition Probabilities}

The average power consumed by a CMOS gate \( n \) is calculated by (1) where

\begin{equation}
E_n(\text{switching}) = \sum_{t \in \mathbb{Z}/n} t_a(t) + t_d(t)
\end{equation}

where \( ES(n) \) denotes the set of probabilistic events for \( n \). We obtain \( E(\text{switching}) \) for every gate in the network in the following manner. We visit the gates in a topological fashion (from primary inputs to primary outputs). When we reach gate \( n \), the set of probabilistic events for its inputs are already known. Using Theorem 3.2, Equations (25) and (26) we calculate the set of probabilistic events at the output of \( n \). The assumption is, of course, that the clock cycle time is long enough for all transitions (including hazards) to settle.

In [4], the transition probabilities are calculated using symbolic simulation which uses the global function of a node (in terms of primary inputs) in order to estimate the transitions at the output of the node. Our method estimates the transition probabilities of a node by closed form formula which only depend on the signal and transition probabilities of the immediate inputs of the node.

Under a fixed binary inertial delay model [8], the gate's inertia is taken into account as follows. If the inertial delay of a gate is non-zero, then all inputs to the gate must remain constant for a period at least equal to the inertial delay. The transition of an input at time \( t \) will not be propagated to the output if there is an opposite transition within the period of the inertial delay from the gate. This is an important consideration, since by ignoring the inertial delay of gates, we may overestimate the dynamic power consumption in the circuit.

\section*{4 Experimental Results}

We compared our power estimation procedure with that of [4] which uses symbolic simulation \(^2\) in terms of accuracy and performance. We applied both algorithms on a subset of the ISCAS-89 benchmark sets. All circuits were mapped by the SIS mapper using lib2-gencil. All primary inputs were assumed to be independent and a 20MHz clock frequency was used. All power estimates are in micro-Watts.

Tables 2 and 3 contain our experimental results showing the average power consumption estimation and run time for a unit delay and a fixed binary inertial delay model, respectively. All run times (in seconds) were obtained on a Sun Sparc 1+ with 64MB of memory. The entry N/A means that results could not be obtained due to excessive memory requirements. It is seen that the run time is sped up by an average factor of 9 while the average error introduced by our approximation is only 1%. For large benchmarks, symbolic simulation cannot complete the estimation.

\section*{5 Concluding Remarks}

We presented an efficient algorithm for average power estimation under real delay model using transition probability propagation. We gave an exact algorithm for uncorrelated inputs and presented an approximate procedure for correlated input using the notion of steady state conditions of transition waveform. Experimental results show that

\(^2\) The memory requirement and run-time of this method have been reduced by using a random logic simulation approach [2].
agate these events through the circuits. When a probabilis- 
tic event occurs at the input to some gate \( n \), the appro- priate event (shifted by an amount equal to the gate delay 
and with corresponding signal and transition probabilities) 
is created at the output of \( n \).

In the following, we show how the signal and transition probabilities at the output of a gate are calculated given the signal and transition probabilities at its inputs. We give an exact method for the case of uncorrelated inputs. In the case of correlated inputs, we present an approxi- mately method which captures the input correlations by us-
ing steady state conditions of the transition waveforms for 
inputs. We use the two-input AND gate as an example.

Similar results can be obtained for the two-input OR gate.\(^1\)

To simplify the presentation, we assume that the pro-
agation delay through the AND gate is pin-independent.

Extension to a pin-dependent gate delay model is straight 
forward.

### 3.1 Two-Input AND Gate with Uncorrelated Inputs

Let \( i_1 \) and \( i_2 \) be the inputs, \( n \) be the output, and \( \delta \) be the propagation delay through the two-input AND gate. Let \( w_{i_1}, w_{i_2} \) and \( w_n \) be the transition waveforms at \( i_1, i_2 \) and \( n \), respectively. \( w_n \) will have a \( \delta \rightarrow t \) transition at time \( t + \delta \) if either \( w_{i_1} \) or \( w_{i_2} \) has a \( \delta \rightarrow t \) transition and the other has a signal value equal to 1 at time \( t \) or both have \( \delta \rightarrow t \) transitions at time \( t \). Hence we can write:

\[
up_{w_n}(t + \delta) = up_{w_{i_1}}(t)sw_{i_2}(t^\circ + \delta)u_{w_{i_2}}(t) 
- up_{w_{i_1}}(t)up_{w_{i_2}}(t). \quad (8)
\]

Since \( i_1 \) and \( i_2 \) are uncorrelated, so are \( w_{i_1} \) and \( w_{i_2} \). Thus,

\[
P_o(w_n) = P_o(w_{i_1} \land w_{i_2}) = P_o(w_{i_1})P_o(w_{i_2}). \quad (9)
\]

Combining (4), (8) and (9) we obtain:

\[
t_{u_n}(t + \delta) = \sum_{w_{i_1} \in W_{i_1}} \sum_{w_{i_2} \in W_{i_2}} \frac{P_o(w_{i_1})P_o(w_{i_2})}{P_o(w_n)} 
\{ up_{w_{i_1}}(t)sw_{i_2}(t^\circ + \delta)u_{w_{i_2}}(t) 
- up_{w_{i_1}}(t)up_{w_{i_2}}(t) \}, \quad (10)
\]

Similarly,

\[
t_{d_n}(t + \delta) = \sum_{w_{i_1} \in W_{i_1}} \sum_{w_{i_2} \in W_{i_2}} \frac{P_o(w_{i_1})P_o(w_{i_2})}{P_o(w_n)} 
\{ down_{w_{i_1}}(t)sw_{i_2}(t^\circ) + sw_{i_1}(t^\circ)down_{w_{i_2}}(t) 
- down_{w_{i_1}}(t)down_{w_{i_2}}(t) \}. \quad (11)
\]

**Theorem 3.1** Given signal and transition probabilities at uncorrelated inputs \( i_1 \) and \( i_2 \) of a two-input AND gate \( n \) with propagation delay \( \delta \), the transition probabilities at the output of \( n \) are calculated as:

\[
t_{u_n}(t + \delta) = tu_{i_1}(t)sp_{i_2}(t^\circ) + sp_{i_1}(t^\circ)tu_{i_2}(t) 
- tu_{i_1}(t)tu_{i_2}(t), \quad (12)
\]

\[
t_{d_n}(t + \delta) = td_{i_1}(t)sp_{i_2}(t^\circ) + sp_{i_1}(t^\circ)td_{i_2}(t) 
- td_{i_1}(t)td_{i_2}(t). \quad (13)
\]

The signal probabilities at the output of \( n \) are given by:

\[
s_{p_n}(t + \delta^+) = sp_{w_n}(t + \delta) + tu_{w_n}(t + \delta) - td_{w_n}(t + \delta) \quad (14)
\]

where

\[
s_{p_n}(t + \delta^-) = sp_{w_n}(t + \delta^+) \quad (15)
\]

and \( t_{\delta} + \delta \) and \( t + \delta \) are two consecutive probabilistic events for \( n \). Note that \( s_{p_n}(t + \delta^+) \) is known from similar calculation performed for the previous event. The calculations are initialized as \( s_{p_n}(0^-) = P(n = 1) \).

### 3.2 Two-input AND Gate with Correlated Inputs

If the input signals are correlated, their corresponding transition waveforms are also correlated. Equation (9) is then replaced by:

\[
P_o(w_n) = P_o(w_{i_1} \land w_{i_2} | w_{i_1}) = P_o(w_{i_1})P_o(w_{i_2} | w_{i_1}). \quad (16)
\]

where \( P_o(w_{i_2} | w_{i_1}) \) is the conditional probability of \( w_{i_2} \) given \( w_{i_1} \). The correlation between two transition waveforms at two nodes depends on the logical waveforms ap-
plicated to the primary inputs and the circuit topology. It is expensive to compute the exact correlation. We instead use the correlation between the steady state conditions of transition waveforms to approximate the actual correlation as detailed next.

We approximate the correlation between two transition waveforms \((w_{i_1}, w_{i_2})\) by assuming that the two waveforms are correlated only by their steady state conditions \((s_{w_{i_1}}, s_{w_{i_2}})\). Thus,

\[
P_o(w_{i_1} \land w_{i_2}) = \frac{P_o(s_{w_{i_1}} \land s_{w_{i_2}})}{P_o(s_{w_{i_1}})P_o(s_{w_{i_2}})}, \quad (17)
\]

and

\[
P_o(w_{i_1} \land w_{i_2}) = \frac{P_o(s_{w_{i_1}} \land s_{w_{i_2}})}{P_o(s_{w_{i_1}})P_o(s_{w_{i_2}})}P_o(w_{i_1})P_o(w_{i_2}). \quad (17)
\]

Let \( s_{w_{i_1}} = (x_1, y_1) \) and \( s_{w_{i_2}} = (x_2, y_2) \). If the primary inputs are uncorrelated and temporally independent, then

\[
P_o(s_{w_{i_1}}) = P(i_1 = x_1)P(i_1 = y_1), \quad (18)
\]

\[
P_o(s_{w_{i_2}}) = P(i_2 = x_2)P(i_2 = y_2), \quad (19)
\]

and

\[
P_o(s_{w_{i_1}} \land s_{w_{i_2}}) = P(i_1 = x_1 \land i_2 = x_2)P(i_1 = y_1 \land i_2 = y_2). \quad (20)
\]

\( P(i_1 \land i_2) \) can be calculated by building an OBDD for the function \( i_1 \land i_2 \) and doing a depth-first traversal on the OBDD to find the signal probability of the function \([5\).]

Each transition waveform is tagged by its steady states \((x, y)\). Similarly, we can tag the signal and transition probabilities. From (2) and (3), we have:

\[
s_{p_n}(t^-) = \sum_{(x, y) \in S} sp_{n}(x, y)(t^-), \quad (21)
\]

\[
s_{p_n}(t^+) = \sum_{(x, y) \in S} sp_{n}(x, y)(t^+), \quad (22)
\]

---

1. Simple gates with \( k > 2 \) inputs are first decomposed into a tree of two-input gates and then processed. Complex gates are first decomposed into a two-level AND-OR representation with inverters. Propagation of transition propagation through an inverter is trivial.

2. Proofs of theorems can be found in [11].
Using equations (2-6), we can define the probability waveform of \( n \) which consists of a sequence of transition edges or events, annotated with signal and transition probabilities. This is similar to the definition given in [6].

Examples of some transition waveforms, the associated probability waveform and some of the definitions given above are shown in Figure 1.

A transition waveform \( w_i \) for gate \( n \) corresponds to one (or more) primary input vector pair(s): \( V_{-1} \) and \( V_0 \). The steady states of \( w_i \) are defined as values at \( t = 0^+ \) (corresponding to the steady state value of \( n \) under \( V_{-1} \)) and \( t = \infty \) (corresponding to steady state value of \( n \) under \( V_0 \)). These states are either \((0,0), (0,1), (1,0) \) or \((1,1) \) (see Figure (1)).

Let \( x, y \in \{0, 1\} \), then

\[
ss_{w_i} : \text{steady state conditions of } w_i, \\
W_n^{(x,y)} : \text{the set of all } w_i \text{'s at } n \text{ such that } ss_{w_i} = (x, y), \\
W_n = W_n^{(0,0)} + W_n^{(0,1)} + W_n^{(1,0)} + W_n^{(1,1)}.
\]

If primary inputs are temporally independent, i.e., the value at input \( i \) at clock cycle \( k+1 \) is independent of its value at clock cycle \( k \), then the sum of the occurrence probabilities of all transition waveforms which have the same steady state conditions \( (x,y) \) is equal to:

\[
\sum_{w_i \in W_n^{(x,y)}} P_o(w_i) = P(n = x)P(n = y) \tag{7}
\]

where \( P(n = x) \) and \( P(n = y) \) denote the probabilities of \( n \) assuming binary values \( x \) and \( y \), respectively.

3 Propagation of Transition Probabilities

Given the set of probabilistic events for each primary input, an event-driven simulation approach is used to prop-
Efficient Estimation of Dynamic Power Consumption under a Real Delay Model

Chi-Ying Tsui, Massoud Pedram, Alvin M. Despain

Department of Electrical Engineering - Systems
University of Southern California, Los Angeles, CA 90089

1 Introduction

With recent advances in microelectronic technology, smaller devices are now possible allowing more functionality on an integrated circuit (IC). Portable applications have shifted from conventional low performance products such as wristwatches and calculators to high throughout and computationally intensive products such as notebook computers and personal digital assistants. The new applications require high speed, yet low power consumption. In addition, power consumption in chips rises with increased integration density and circuit speed. Low power design is thus essential to lower the packaging and cooling costs and prolong the life of the ICs. To achieve this goal, designers are willing to trade off area and performance for low power consumption.

Recently a number of researchers have addressed the problem of reducing power consumption during logic synthesis and physical design [9][12][7][10][13]. All of these approaches are based on a zero delay timing model. This model however underestimates the actual power consumption of the circuit as it ignores the glitches which contribute significantly to the total power consumption.

Several researchers have studied the problem of estimating power consumption under a real delay model. Najm et al. [6] use a probabilistic simulation approach to estimate the average current drawn by a circuit. Their approach is built on the notion of transition probabilities over a time period. They define the transition probability of a signal \( N \) at time \( t \) as the probability of a low-to-high (or high-to-low) transition from \( t' \) (just before \( t \)) to \( t'' \) (just after \( t \)). Given transition probabilities (over time) at the circuit inputs, the transition probabilities (over time) at internal nodes are derived by propagating transition probabilities through nodes using an approach often employed by event-driven timing simulators. The main shortcoming of this approach is that during the propagation step, inputs to the gates (nodes) are assumed to be independent.

Ghosh et al. [4] use symbolic simulation in order to produce a set of Boolean functions which represent conditions for switching at each gate in the circuit at a specific time instance. Given input switching rates, the transition probability at each gate is calculated by performing a linear traversal of the Ordered Binary Decision Diagrams (OBDDs) [1] representation of the Boolean function constructed for the given time instance [5]. A real delay model which correctly computes the Boolean conditions that cause glitching is used and correlations due to the re-convergence of input signals are taken into account. This procedure is exact, however, suffers from excessive computation time and storage space.

In this paper, we present a fast and memory efficient power estimation method using the notion of transition probabilities introduced in [6]. We describe a linear time algorithm which propagates the transition probabilities from the circuit inputs to the circuit outputs. For the case in which inputs to the gates are uncorrelated, our method is similar to [6]. We use the notion of transition waveforms to prove that this propagation mechanism is correct and exact. We then extend the method to account for input correlations by proposing a novel tagging mechanism which uses the notion of steady state conditions of the transition waveforms. This extension, although only approximately captures the input correlations during the transition probability propagation step, is very accurate (as confirmed by our empirical data) and is valid under the most general delay models.

The rest of this paper is organized as follows. In Section 2 we review the notions of signal and transition probabilities and describe a model for power consumption. Propagation of transition probabilities through the network is discussed in Section 3. In particular, we describe a novel tagging mechanism based on the steady state conditions of the input waveforms that allow us to capture the input correlations. Experimental results and conclusions are presented in Sections 4 and 5 respectively.

2 Background and Terminology

The average power consumption of a CMOS gate \( n \) in a synchronous CMOS circuit is given by:

\[
P_{avg} = 0.5 \frac{V^2}{T_{cycle}} C_{load} E_n(switching)
\]  

(1)