Estimation of Average Switching Activity in Combinational and Sequential Circuits

Abhijit Ghosh
Mitsubishi Electronics America

Srinivas Devadas
MIT, Cambridge

Kurt Keutzer
Synopsys

Jacob White
MIT, Cambridge

Abstract

We address the problem of estimating the average power dissipated in VLSI combinational and sequential circuits, under random input sequences. Switching activity is strongly affected by gate delays and for this reason we use a general delay model in estimating switching activity. Our method takes into account correlation caused at internal gates in the circuit due to reconvergence of input signals. In sequential circuits, the input sequence applied to the combinational portion of the circuit is highly correlated because some of the inputs to the combinational logic are flip-flop outputs representing the state of the circuit. We present methods to probabilistically estimate switching activity in sequential circuits. These methods automatically compute the switching rates and correlations between flip-flop outputs.

1 Introduction

We address the problem of estimating the average power dissipated in combinational and sequential VLSI circuits given random input sequences. This measure can be used to make architectural or design-style decisions during the VLSI synthesis process.

Probabilistic methods for power or current estimation are attractive because statistical estimates can be obtained without recourse to time-consuming exhaustive simulation. In the past, probabilistic peak current estimation methods (e.g., [5]) that compute probabilistic current waveforms for combinational circuits have been developed. Estimation of worst-case power dissipation (e.g., [10], [7]) is a difficult problem requiring a branch-and-bound search and these methods have been applied to small to moderate sized circuits.

The problem of estimating average power in combinational circuits can be reduced to one of computing signal probabilities [1] of a multilevel circuit derived from the original circuit by a process of symbolic simulation. The work closest to our own is the work on transition density calculation by Najm [11]. Transition densities correspond to average switching rates for gates in the circuit. In [11], an interconnection of combinational logic modules, each with a certain delay, makes up a circuit. Transition densities are propagated through combinational logic modules without regard to their structure. Correlations between internal lines due to reconvergence are ignored during propagation. It is possible to take into account correlations by lumping all the modules into one large module, but in this case the information regarding the delay of the individual modules is lost. Cirit in [8] gives methods to calculate dynamic power dissipation based on approximate signal probability evaluation procedures.

Our work improves upon the state-of-the-art in several ways. We use a general delay model for combinational logic in our symbolic simulation method, which correctly computes the Boolean conditions that cause *glitching* (multiple transitions at a gate) in the circuit. In some cases, glitching may account for a significant percentage of the dissipated power or switching activity. Symbolic simulation produces a set of Boolean functions that represent the conditions for switching at different time points, for each gate in the circuit. Given input switching rates, we can use various exact or approximate methods to compute the probability of each gate switching at any particular time point. We then sum these probabilities over all the gates to obtain the expected switching activity in the entire circuit over all the time points corresponding to a clock cycle. Our method takes into account correlation caused at internal gates in the circuit due to reconvergence of input signals (reconvergent fanout).

Most VLSI circuits are sequential. We make an important extension to our estimation algorithms to handle sequential circuits. The combinational part of the circuit receives primary inputs as well as inputs from flip-flop outputs. Given the pin restrictions on VLSI chips, the number of flip-flops in large sequential circuits can be many times the number of primary inputs. Previous methods for power estimation assume uniform switching rates for all the inputs to the combinational logic or expect that switching rates are pre-computed using methods like random simulation. Further, the correlation [1] between the flip-flop outputs between one time frame and the next is ignored. Using inaccurate switching rates and ignoring this correlation can cause substantial errors in power estimation, especially for datapath circuits. We develop a method that *automatically computes the switching rates* for each of the flip-flop inputs, *taking into account this correlation between the flip-flops.*

---

[1]For example, flip-flop A may only switch from 0 → 1 when flip-flop B switches from 1 → 0.

253

2 Preliminaries

2.1 A Power Dissipation Model

The amount of energy dissipated by a CMOS logic gate each time its output changes is roughly equal to the change in energy stored in the gate's output capacitance. If the gate is part of a synchronous digital system controlled by a global clock, it follows that the average power dissipated by the gate is given by:

$$P_{avg} = 0.5 \times C_{load} \times (V_{dd}^2/T_{cyc}) \times E(transitions) \quad (1)$$

where P_{avg} denotes the average power, C_{load} is the load capacitance, V_{dd} is the supply voltage, T_{cyc} is the global clock period, and $E(transitions)$ is the *expected value* of the number of gate output transitions per global clock cycle[11], or equivalently the average number of gate output transitions per clock cycle. All of the parameters in (1) can be determined from technology or circuit layout information except $E(transitions)$, which depends on both the logic function being performed and the statistical properties of the primary inputs.

2.2 Static Probabilities

Consider the case of dynamic CMOS logic (e.g. Domino). At the beginning of each clock cycle, all the gates are precharged, and gates make transitions only if their associated Boolean functions are satisfied. For example, a three-input AND-OR gate's Boolean function might be

$$(i_1 \cdot i_2) \vee (i_2 \cdot i_3), \quad (2)$$

where i_1, i_2, and i_3 are primary inputs. In this case, the expected value of the number of transitions at this gate's output is

$$E(transitions) = 2 \times P((i_1 \cdot i_2) \vee (i_2 \cdot i_3) = 1) \quad (3)$$

where $P(x)$ is defined as probability that x is true, and the factor of two in the equation accounts for the reset transition during precharge.

To evaluate (3), it is necessary to determine the primary input probabilities. We assume that *primary inputs are uncorrelated*, and that each is a waveform in time whose value is either zero or one, changing instantaneously at global clock edges. Assuming ergodicity without further comment, the probability of a particular input i_j being one at a given point in time, denoted $p_j{}^{one}$, is given by

$$p_j{}^{one} = \lim_{N \to \infty} \frac{\sum_{k=1}^{N} i_j(k)}{N} \quad (4)$$

where N is the total number of global clock cycles and $i_j(k)$ is the value of input i_j during clock cycle k. Clearly, the probability that i_j is zero at a given point in time, denoted $p_j{}^{zero}$ is

$$p_j{}^{zero} = 1 - p_j{}^{one}$$

We refer to $p_j{}^{zero}$ and $p_j{}^{one}$ as static probabilities. Note that

$$P((i_1 \cdot i_2) \vee (i_2 \cdot i_3) = 1) \neq p_1^{one} p_2^{one} + p_2^{one} p_3^{one}$$

because the first and second product terms are not independent. Rather,

$$P((i_1 \cdot i_2) \vee (i_2 \cdot i_3) = 1) =$$

$$P((i_1 \cdot i_2) \vee (\overline{i_1} \cdot i_2 \cdot i_3) = 1) = p_1^{one} p_2^{one} + p_1^{zero} p_2^{one} p_3^{one}$$

where the second equality holds because $i_1 \cdot i_2$ is disjoint from $\overline{i_1} \cdot i_2 \cdot i_3$. In this example, $(i_1 \cdot i_2) \vee (\overline{i_1} \cdot i_2 \cdot i_3)$ represents a disjoint cover for the logic function, and the terms $i_1 \cdot i_2$ and $\overline{i_1} \cdot i_2 \cdot i_3$ are referred to as *cubes* in the cover[2]. The equivalent logical expression, $(i_1 \cdot i_2) \vee (i_2 \cdot i_3)$, does not represent a disjoint cover because $i_1 \cdot i_2 \cdot i_3$ is contained in both cubes $i_1 \cdot i_2$ and $i_2 \cdot i_3$.

In general, given a disjoint cover for a Boolean function of uncorrelated inputs described by static probabilities, it is easy to determine the probability of the function evaluating to a 1. The procedure is given in the following two theorems whose proof follows directly from elementary probability[12].

Theorem 2.1 : *Given any disjoint cover for a Boolean function, the probability of the function evaluating to a 1 is equal to the sum of the probabilities of each of the cubes in the cover evaluating to a 1.*

Theorem 2.2 : *Given a logical function of uncorrelated inputs in the form*

$$f = i_{\alpha_1} \cdot i_{\alpha_2} \cdot \dots \cdot i_{\alpha_M} \cdot \overline{i_{\beta_1}} \cdot \overline{i_{\beta_2}} \cdot \dots \cdot \overline{i_{\beta_N}}$$

where the i_{α_j}'s are nonnegated inputs and the $\overline{i_{\beta_k}}$'s are negated inputs, then

$$P(f = 1) = p_{\alpha_1}^{one} \cdot p_{\alpha_2}^{one} \cdot \dots \cdot p_{\alpha_M}^{one} \cdot p_{\beta_1}^{zero} \cdot p_{\beta_2}^{zero} \cdot \dots \cdot p_{\beta_N}^{zero}$$

2.3 Transition Probabilities

For static combinational CMOS logic, a gate output can only change when its inputs change, and then only if the Boolean function describing the gate evaluates differently. For example, a 2-input AND gate's output will change between clock cycle t and $t + 1$ if

$$(i_1(t) \cdot i_2(t)) \oplus (i_1(t + 1) \cdot i_2(t + 1)) \quad (5)$$

evaluates to 1, where $i_1(t)$, $i_2(t)$ and $i_1(t+1)$, $i_2(t+1)$ are the inputs to the gate at clock cycle t and $t + 1$ respectively. The disjoint cover for (5) is

$$(i_1(t) \cdot i_2(t)) \cdot \overline{(i_1(t + 1)} \vee (i_1(t) \cdot i_2(t)) \cdot (i_1(t+1) \cdot \overline{i_2(t + 1)}) \vee$$

$$\overline{i_1(t)} \cdot (i_1(t+1) \cdot i_2(t+1)) \vee (i_1(t) \cdot \overline{i_2(t)})(i_1(t+1) \cdot i_2(t+1)). \quad (6)$$

It is not possible to use Theorem (2.2) to evaluate the probability of (6), because an input at time $t+1$ is correlated to its behavior at time t (as in a sequential circuit). Instead, transition probabilities for the transitions $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$ and $1 \rightarrow 1$ must be used. We denote these transition probabilities by p_i^{00}, p_i^{01}, p_i^{10}, and p_i^{11}, respectively, where for example p_i^{10} is defined by:

$$p_j^{10} = \lim_{N \to \infty} \frac{\sum_{k=1}^{N} i_j(k)\overline{i_j(k+1)}}{N} \quad (7)$$

The other transition probabilities follow similarly.

Static probabilities can be computed from transition probabilities, but *not* vice-versa, because of correlation between one time frame and the next. So in general it is necessary to specify transition probabilities. The relations between static probabilities and transition probabilities follow directly from the definitions in (4) and (7), specifically,

$$p_j^{zero} = p_j^{00} + p_j^{01} \quad (8)$$

$$p_j^{one} = p_j^{11} + p_j^{10}$$

Both static and transition probabilities are used to compute $E(transitions)$ for static logic circuits, as can be seen from the expression for the probability that (6) evaluates to a 1,

$$p_1^{10} \cdot p_2^{one} + p_1^{11} \cdot p_2^{10} + p_1^{01} \cdot p_2^{one} + p_1^{11} \cdot p_2^{01} \quad (9)$$

which for this example is also $E(transitions)$. For all primary inputs, it may be assumed that successive input vectors are uncorrelated and a 1 or a 0 are equally likely. Therefore, all transition probabilities may be be assumed to be 0.25, and all static probabilities to be 0.5.

2.4 General Combinational Networks

The algorithm for computing the average power dissipated in a dynamic CMOS combinational network follows directly from the approach in Section (2.2). That is, for each gate g_i in a logical network, we first determine the Boolean function of the gate, f_i, in terms of the network's primary inputs. Then we find a disjoint cover for f_i, and use Theorem (2.1) and (2.2) to evaluate $P(f_i = 1)$.

The average power dissipation for the network is then

$$P_{avg} = (V_{dd}^2/T_{cyc}) \times \sum C_i \times E(transitions \; of \; g_i)$$

$$= (V_{dd}^2/T_{cyc}) \times \sum C_i \times P(f_i = 1) \quad (10)$$

where C_i is the load capacitance of the ith gate and the summation is over all gates in the circuit. Note that (10) follows from the fact that the average value of a sum is equal to the sum of the average values, regardless of correlation[12].
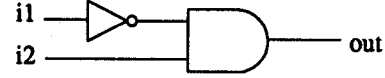

Figure 1: Glitching in a Static CMOS Circuit

A similar overall approach can be used to compute the average power dissipated in a static CMOS combinational network. However, as described in Section (2.3), a two-vector input sequence is required to stimulate activity in static gates. Therefore, the Boolean function for static gate output activity is different than that for a dynamic gate, and the probability of the function being satisfied requires transition probabilities for evaluation. In particular, assuming negligible gate delays, a static CMOS combinational logic gate's output will switch with a change in the primary input vector from $V0$ to Vt if:

$$f_i = ((h_i(V0) = 0) \wedge (h_i(Vt) = 1)) \bigvee$$
$$((h_i(V0) = 1) \wedge (h_i(Vt) = 0)) \quad (11)$$

is satisfied, where h_i is the logic function corresponding to gate g_i's output.

The average power dissipated in the static network is then computed by using (10), with f_i being given by (11), and $P(f_i = 1)$ is evaluated using both transition and static probabilities, as described in Section (2.3).

Instead of enumerating disjoint covers, Binary Decision Diagrams (BDDs) [3] can be used for the calculation of signal probability. It has been shown in [6] and [11] that exact signal probability calculation for a given function can be performed by a linear traversal of a BDD representation of a logic function. We have implemented methods for signal probability calculation using BDDs.

3 Gate Delay Effects

As mentioned above, for static CMOS circuits, switching activity must be analyzed based on considering an input vector pair, denoted $< V0, \; Vt >$. If the gates have appreciable delays, there may be output glitches that can contribute significantly to dissipated power.

For instance, consider the circuit of Figure 1. Assuming that the delays of the inverter and the AND gate are both 1 time unit, if we apply the vector $i_1 = 0$, $i_2 = 0$, followed by $i_1 = 1$, $i_2 = 1$, we will obtain a glitch at the output *out*, which could cause power dissipation. Below, we present a symbolic simulation method that can be used to generate a multiple-output function that represents total switching activity over any possible input vector pair, assuming a general delay model for the gates in the circuit.

3.1 Symbolic Simulation

3.1.1 An Example

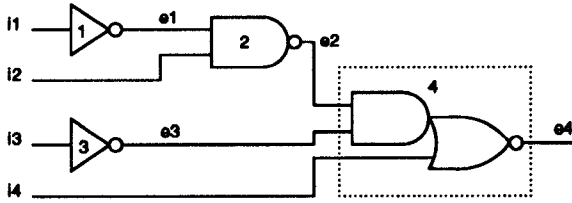Consider the multilevel combinational circuit shown in Figure 2.
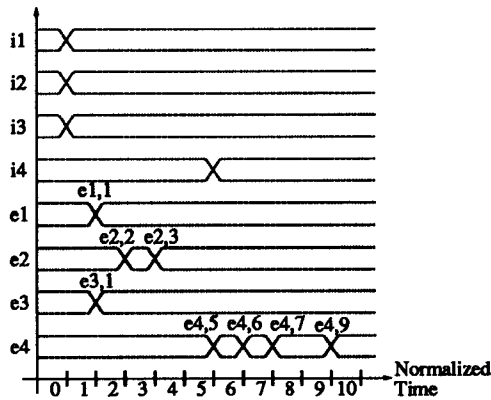
Figure 2: A multilevel combinational circuit

Figure 3: Signal waveforms for the primary inputs and the outputs of the logic gates

It has four primary inputs and one output and consists of four CMOS gates. Figure 3 shows the signal transitions at the primary input nodes as well as the possible transitions at the internal nodes of the network. The time points are in normalized units. The first three inputs, i_1, i_2, and i_3, switch simultaneously between time periods 0 and 1. The fourth input, i_4, is a late arriving signal that switches between the time points 5 and 6. In this example, the delays of both Gate 1 and Gate 3 are one time unit. Gate 2 has a delay of two units while Gate 4 has a delay of four units.

Also shown in Figure 3 are the waveforms, e_i's, representing the signals at the outputs of ith logic gates. Each of the possible transitions, $e_{i,j}$, represents either a low-to-high or high-to-low signal transition between $[j]^{th}$ and $[j+1]^{th}$ time points. The number of all possible transitions at a gate output may equal the sum of all possible transitions at the gate inputs. These transitions are delayed by the gate's propagation delay.

3.1.2 Unit-Delay Model

Even under the idealization of a unit-delay model, the gate output nodes of a multilevel network can have multiple transitions in response to a two-vector input sequence. In fact, it is possible for a gate output to have as many transitions as levels in the network.

We construct the Boolean functions describing the gate outputs at the discrete points in time implied by the unit-delay model. That is, we consider only discrete times t, $t + 1$, ..., $t + l$, where t is the time when the inputs change from $V0$ to Vt, and l is the number of levels in the network. For each gate output i, we use symbolic simulation to construct the $l + 1$ Boolean functions $f_i(t+j)$, $j \in 0$, ..., l which evaluate to 1 if the gate's output is 1 at time $t+j$. Note that as we assume no gate has zero delay and that the network has settled before the inputs are changed from $V0$ to Vt, $f_i(t)$ is the logic function performed for $V0$ at the ith gate output. Finally, we can determine whether a transition occurs at a boundary between discrete time intervals $t+j$ and $t+j+1$ by exclusive-OR'ing $f_i(t+j)$ with $f_i(t + j + 1)$.

3.1.3 General Delay Model

A gate with a large fanin may have several times the delay of an inverter. If one uses normalized time units, one can always introduce unit-delay buffers at the output of gates in a circuit, which have a delay greater than unity, in order to model differing delays among logic gates.

Further, all gates have *inertial delays*, i.e., all gates require energy to switch their output nodes. For a transition at an input terminal to propagate through the gate, the state of the input signal before and after the transition must persist for a minimum duration equal to the delay of the gate. Any high frequency pulse at the inputs whose pulse duration is shorter than the gate delay (or some fraction of the gate delay) will be filtered out.

Our general symbolic simulator is able to simulate circuits with arbitrary gate transport and inertial delays (without introducing any unit delay buffers in the circuit). The simulator processes one gate at a time, moving from the primary inputs to the primary outputs of the circuit. For each gate, the possible transition times of its inputs are first obtained. Then, possible transitions at the output of the gate is derived, taking into account transport and inertial delays. If the inertial delay of a gate is non-zero, then all inputs to the gate must remain the same for a period equal to the inertial delay (or a fraction of it) for the output of the gate to make a transition.

4 Sequential Circuits

4.1 Introduction and Motivation

VLSI circuits are sequential, i.e., they contain memory elements or flip-flops. This sequential nature of VLSI circuits makes the estimation of power dissipation more complicated than for combinational circuits.

A common model for a sequential circuit is shown in Figure 4. We will denote the vector pair applied to the combinational logic as $< V0, Vt >$. $V0$ and Vt have a primary input part and a present-state part. $V0$ is denoted $I0@P0$ and Vt is denoted $It@Pt$, where $I0$ and It correspond to the primary input parts, and $P0$ and Pt correspond to the present-state parts.
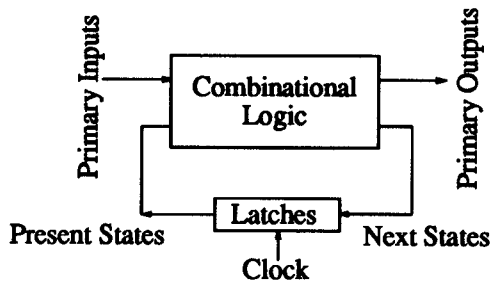
Figure 4: A General Synchronous Sequential Circuit
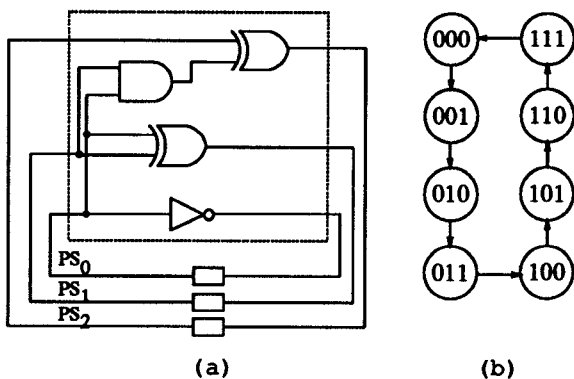

(a)   (b)

Figure 5: A 8-State Counter

One can ignore the feedback corresponding to the next-state lines and present-state lines, and estimate the power dissipated by the combinational logic using the methods presented in the previous sections. This strategy is a relatively crude approximation for two reasons as we will illustrate with a simple example.

Consider the STG of Figure 5(b). It represents the behavior of an 8-state autonomous counter, whose logic implementation is shown in Figure 5(a). Assuming that the counter can begin from any state with uniform probability, the probability of line PS_2 making a $1 \to 0$ transition is 0.125, and probabilities for $0 \to 1$, $0 \to 0$ and $1 \to 0$ transitions are 0.125, 0.375 and 0.375, respectively. On the other hand, the probability of the line PS_0 making a $1 \to 0$ transition is 0.5, and the probabilities for a $0 \to 1$, $0 \to 0$ and $1 \to 1$ are 0.5, 0, and 0, respectively. The transition probabilities for the different present-state lines are different.

To make matters worse, the transitions on one present-state line are correlated to the transitions on the other present-state lines. For instance, in Figure 5, it is easy to see that PS_2 makes a $0 \to 1$ transition only when PS_0 makes a $1 \to 0$ transition. Simply computing the correct switching rates does not necessarily result in exact power dissipation estimation if we ignore this correlation of the transitions. Note however, that in this example, the present-state lines are uncorrelated, $i.e.$, given the static probability of each present-state line, the probability of a particular state can be computed correctly.
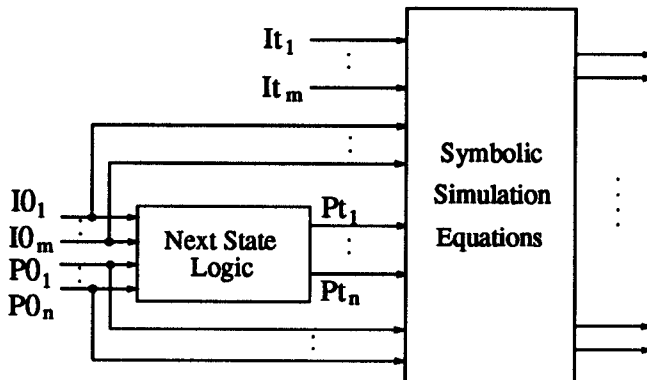

Figure 6: Taking Correlation Into Account

4.2 Power Estimation Method

We will assume that the circuit, upon power-up, can begin in any of the 2^n states, where n is the number of flip-flops. Further, we assume that the STG of the machine is strongly connected, and that an arbitrary number of cycles after power-up, the probability that the machine is in any of its 2^n states is uniform.

Consider the circuit of Figure 6. It has two blocks: the first is the combinational logic implementing the next-state function of the given machine, and is used to derive the next state from the present state. The first block feeds a second block, which represents the Boolean function obtained by symbolic simulation of the combinational logic of the machine.

The decomposition of Figure 6 implies that the gate output switching activity can be determined given the vector pair $< I0, It >$ for the primary inputs, but only $P0$ for the states. Therefore, to compute gate output transition probabilities, we require the transition probabilities for the primary inputs, but only require the static probabilities for the present state. This use of the next-state logic generates Boolean equations which model the correlation between the present and the next states, whereby the transition probabilities are automatically computed, taking into account the correlation between transitions. However, to use the average power estimation techniques described in Section 2, we must assume that the present-state lines are uncorrelated. We discuss the validity of this assumption in the next section.

4.3 STG Characteristics

The state bits in a synchronous sequential circuit are uncorrelated if the machine is equally likely to be in any of its 2^n states an arbitrary number of cycles after power-up (in which case the static probability of each present-state line is 0.5). There are many reasons why this might not be true. In particular if the entire state space is not strongly connected, the machine might operate only in a strongly connected portion of the state space. In general, after an arbitrarily long period of time, the machine will be in a strongly connected portion of its STG, which might contain just

a single state. If the assumption above is not valid, then static probability of 0.5 cannot be assumed for the present-state lines. Moreover, because of the correlation between the present-state lines, the power estimate obtained is approximate.

Given a set of states in a strongly connected portion of the STG in which the machine can be, the probability of being in any particular state within the set is not necessarily uniform. In particular, it might depend on the *in-degree*[2] of a state. States with a large in-degree may have greater probabilities of being entered. The exact probabilities of being in particular states can be derived from the Chapman-Kolmogorov equations for discrete-time Markov Chains [12]. However, the number of equations will be exponential in the number of latches and inputs in the circuit, and therefore cannot be solved for most practical circuits.

In practice, large datapath circuits satisfy the assumptions because of their highly connected STGs that include almost all of 2^n states, and states with similar in-degrees. Thus, the assumption made regarding the structure of the STG is quite realistic for such circuits. However, controllers might violate these assumptions. In that case, because of the correlation between present-state lines, we will only be able to obtain an approximate power estimate. We can increase the accuracy of the power estimate by calculating the static probabilities of the present-state lines more accurately. Techniques to compute the reachable set of states, or the strongly connected portion of the STG of a sequential machine, can be developed, based on the strategies of [9, 4]. These techniques are viable for machines with up to approximately 50 flip-flops. Thereafter, the static probability of each present-state line can be calculated. For example, if a machine has states 00, 01, and 10, and it is in these states with proabilities 0.1, 0.5, 0.4, respectively, then the probability of the first present-state line being 1 is 0.4 and that of the second is 0.5. Finally, during signal probability calculation, we have to exclude the set of unreachable states. This is performed by computing the logical AND of the BDD representing the set of reachable states with the BDD representing the switching function.

5 Experimental Results

Throughout this section, we will be measuring the average power dissipation of the circuit by using (1) summed over all the gates in the circuit. The N_G values are computed for the gates in the circuit under different delay models. The capacitance values of the gates are assumed to be given, or can be computed using the fanin and fanout numbers and the size (in terms of the number of transistors) for each gate, or can be obtained after mapping.

The statistics of the examples used are shown in Table 1. All of the examples except the last two belong to the ISCAS-89 Sequential Benchmark set. Example **add16** is a 16-bit adder and accumulator, where the

---

[2]The number of edges in the STG that enter this state.

CKT	Inputs	Outputs	Latches	Gates
s27	4	1	3	10
s298	3	6	14	119
s349	9	11	15	150
s386	7	7	6	159
s420	19	2	16	196
s510	19	7	6	211
s641	35	24	19	379
s713	35	23	19	393
s838	35	2	32	390
s1238	14	14	18	508
s1494	8	19	6	647
s5378	35	49	179	2779
add16	33	17	16	288
max16	33	16	16	154

Table 1: Statistics of examples

CKT	Combinational		Sequential	
	Power	Time	Power	Time
s27	14	0.1	15	0.1
s298	212	0.9	259	1.8
s349	280	1.6	258	2.9
s386	264	1.5	428	2.1
s420	287	1.8	336	2.8
s510	425	2.2	425	3.7
s641	582	3.5	580	6.7
s713	650	4.3	649	6.3
s838	568	4.4	672	4.6
s1238	914	6.2	904	6.4
s1494	1002	8.3	1185	13.8
s5378	4752	11.0	4823	45.1
add16	58	0.4	58	0.6
max16	56	0.4	73	0.7

Table 2: Power estimation for dynamic circuits

output of the adder is fed back to one set of inputs. Example **max16** is similar, with the adder replaced by maximum finder. Some of these circuits have over 200 inputs and outputs (*e.g.*, **s5378**). All the circuits considered are technology mapped circuits.

If the circuits are treated as purely combinational dynamic circuits, then the average-case power dissipation is shown in the second column of Table 2, while the time required to obtain the power dissipation is shown in the third column. These times have been obtained on a DEC-station 5900 with 440Mb of memory, and are in seconds. The power estimates, in microWatts, have been obtained assuming a clock frequency of 20MHz. Treating the circuit as a dynamic sequential circuit, the average power dissipation and the time required to derive the power dissipation are shown in columns four and five respectively of Table 2. Note that taking correlations of state transitions into account, as described in Section 4.2, produces significant differences in the power estimate.

For the remainder of this section, we consider the circuits to be static CMOS circuits. In Table 3, we treat the circuits as purely combinational and show the effects of various delay models on the power estimate. In the zero delay model, all gates have zero delay and therefore they switch instantaneously. In the unit delay model, all gates have one unit delay. Using the zero delay model ignores glitches in the circuit, and therefore power dissipation due to glitches are not

CKT	Zero Delay Power	Unit Delay Power	Variable Delay Power	Variable Delay Time
s27	15	19	17	0.1
s298	190	215	198	1.6
s349	238	358	308	4.7
s386	252	288	271	3.8
s420	108	137	128	8.2
s510	271	346	304	3.1
s641	481	665	596	41.0
s713	513	669	637	45.1
s838	184	253	214	49.3
s1238	622	790	716	31.8
s1494	1032	1280	1194	20.0
s5378	3677	4513	4099	2022.2
add16	58	102	102	8.1
max16	56	85	85	6.4

Table 3: Power estimation for combinational circuits

CKT	Zero Delay Power	Unit Delay Power	Variable Delay Power	Variable Delay Time
s27	11	15	14	0.2
s298	168	193	173	5.2
s349	225	275	262	6.5
s386	300	352	333	5.4
s420	126	152	150	10.5
s510	211	265	237	7.8
s641	399	539	493	40.4
s713	429	537	528	50.1
s838	230	278	275	62.4
s1238	613	781	706	45.0
s1494	1072	1498	1303	45.1
s5378	3644	4620	3978	226.8
add16	71	129	129	11.3
max16	46	75	75	8.5

Table 4: Power estimation for sequential circuits

taken into account. The unit delay model takes into account glitches, but does not take the inertial delay of gates into account and therefore may overestimate the power dissipation. The variable delay model with inertial gate delays is the most realistic one, and its estimates, as expected, lie between the zero delay and the unit delay model. Only the times required to obtain the power estimate for the variable delay model is shown in the last column.

The results obtained by treating the circuits as sequential, using the method of Section 4.2, are shown in Table 4. While considering these circuits as sequential, we have taken into account the correlation between two successive values on a present-state line using the method of Section 4.2. This again results in significant differences in the power estimate, as can be seen by comparing Tables 3 and 4.

For each circuit, we compared the power estimates obtained by our method to that obtained by simulating the circuit. The power estimate was obtained by logic simulation, until convergence, with randomly generated input sequence applied to the sequential circuit. The same power dissipation model was used. We obtained excellent agreement (to within 2%) between the power estimate using our method and the random simulation estimate. However, power estimation by simulation took anywhere between 2 to 100 times longer.

References

[1] P. H. Bardell, W. H. McAnney, and J. Savir. *Built-In Test for VLSI: Pseudorandom Techniques.* John Wiley and Sons, New York, New York, 1987.

[2] R. Brayton, G. Hachtel, C. McMullen, and A. Sangiovanni-Vincentelli. *Logic Minimization Algorithms for VLSI Synthesis.* Kluwer Academic Publishers, 1984.

[3] R. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. In *IEEE Transactions on Computers,* volume C-35, pages 677–691, August 1986.

[4] J. Burch, E. Clarke, K. McMillan, and D. Dill. Sequential Circuit Verification Using Symbolic Model Checking. In *Proceedings of the 27th Design Automation Conference,* pages 46–51, June 1990.

[5] R. Burch, F. Najm, P. Yang, and D. Hocevar. Pattern-independent Current Estimation for Reliability Analysis of CMOS Circuits. In *Proceedings of the 25th Design Automation Conference,* pages 294–299, June 1988.

[6] S. Chakravarty. On the Complexity of Using BDDs for the Synthesis and Analysis of Boolean Circuits. In *Proceedings of the 27th Annual Allerton Conference on Communication, Control, and Computing,* pages 730–739, September 1989.

[7] S. Chowdhury and J. S. Barkatullah. Estimation of Maximum Currents in MOS IC Logic Circuits. In *IEEE Transactions on Computer-Aided Design,* volume 9, pages 642–654, June 1990.

[8] M. A. Cirit. Estimating Dynamic Power Consumption of CMOS Circuits. In *Proceedings of the Int'l Conference on Computer-Aided Design,* pages 534–537, November 1987.

[9] O. Coudert, C. Berthet, and J. C. Madre. Verification of Sequential Machines Using Boolean Functional Vectors. In *IMEC-IFIP Int'l Workshop on Applied Formal Methods for Correct VLSI Design,* pages 111–128, November 1989.

[10] S. Devadas, K. Keutzer, and J. White. Estimation of Power Dissipation in CMOS Combinational Circuits. In *Proceedings of the Custom Integrated Circuits Conference,* pages 19.7.1–19.7.6, May 1990.

[11] F. Najm. Transition Density, A Stochastic Measure of Activity in Digital Circuits. In *Proceedings of the 28th Design Automation Conference,* pages 644–649, June 1991.

[12] A. Papoulis. *Probability, Random Variables and Stochastic Processes.* McGraw Hill, 1984.