Implicit Resolution of the Chapman-Kolmogorov Equations for Sequential Circuits: An Application in Power Estimation

Ana T. Freitas
atf@inesc.pt
IST-INESC
Lisbon, Portugal

Arlindo L. Oliveira
aml@inesc.pt
IST-INESC/CEL
Lisbon, Portugal

Abstract

In this work we describe an approach that implicitly formulates and solves the Chapman-Kolmogorov equations that describe the state probabilities associated with the stationary behavior of sequential circuits. Unlike previous approaches that assumed uncorrelated input signals, we model the more general case where the sequential circuit is driven by a sequence of inputs described by a discrete time Markov chain. This Markov chain is described implicitly using a formalism that allows for a compact description of chains with an exponentially high number of states. Using this approach, we present an application in power estimation of sequential circuits that takes into account all the temporal and spatial correlations between the primary inputs and the internal signals. We present results showing that, in some cases, it is possible to solve exactly the Chapman-Kolmogorov equations for systems with more than 10^7 *equations.*

1 Introduction and related work

The behavior of a finite state machine driven by a sequence of inputs that follows a probabilistic distribution subject to some general constraints can be studied by viewing its transition graph as a Markov chain.

Many algorithm have been proposed to analyze Markov chains of varying degrees of complexity. Advanced numerical techniques have been applied to systems with transition structures of varying sizes, but any approach that needs to explicitly list all the transitions in a Markov chain is necessarily limited in the size of the systems it can handle.

For this reason, other authors have proposed symbolic algorithms that compute the steady-state probabilities for very large finite state machines [4]. These algorithms, based on ADD's, determine the steady-state probabilities by regarding finite state machines as homogeneous, discrete-parameter Markov chains with finite state spaces, and by solving the corresponding Chapman-Kolmogorov equations. However, with this approach is only possible to model systems when successive values of the primary inputs are considered independent. This restriction is too strong for many realistic applications.

The approach proposed in this work considers that the primary inputs are modeled by a discrete time Markov chain. The method proposed in this work computes the composed Markov chain [5] that results from the interaction of the input probability distribution with the sequential behavior of the circuit dictated by its state transition graph.

An important application of methods that solve the equations that govern the steady state behavior of sequential circuits is power estimation. When one wants to perform power estimation of sequential circuits, it is important to consider the power dissipated when the circuit reaches the steady state. In fact, the internal signal probabilities in the steady state affect very strongly the value computed for the dissipated power and can be obtained from the Chapman-Kolmogorov equations of the circuit.

In this context, we show that the approach can be effectively applied to probabilistic power estimation, by using an algorithm [2] that estimates the switching activity in circuit nodes given the Markov chain that governs the distribution of the input signals.

The first probabilistic method that estimates switching activity at the logic level in sequential circuits was presented in [3]. This method can accurately model the correlation between the applied vector pairs, but assumes that the state probabilities are all uniform. A more accurate probabilistic approach was suggested in [8]. This method handles the sequential nature of the circuit by solving the Chapman-Kolmogorov equations. However, this method does not take into account the correlations between the primary inputs and the present state lines.

Using the probabilistic power estimation approach described in this work it is possible to compute the power dissipated in sequential circuits taking into account all the temporal and spatial correlations between the signals.

1530-1591/03 $17.00 © 2003 IEEE

2 Modeling the Markov chain at the inputs

The modeling of the Markov chain at the inputs used in this work has been proposed in [2] and is based on the definition of a set of Input Correlations Functions (ICF), $\mathcal{F} = \{(F_1, P_1), (F_2, P_2), \ldots, (F_r, P_r)\}$, that describes the correlations at the inputs and associated probabilities. This set of functions describes a discrete-time Markov chain by specifying which transitions are possible at the primary inputs of the circuit. If there are n input nodes, this set describes, in a compact way, a discrete-time Markov chain with $K = 2^n$ states, one for each possible input combination.

A discrete-time Markov chain is a Markov process $\{X^t \mid t \in T\}$ with a finite number of states $A = \{a_1, \ldots, a_K\}$, [6]. We assume that the space T is discrete. Such a process is specified in terms of its state probabilities

$$P\{X^t = a_i\} \;\; i = 1, 2, \ldots K \tag{1}$$

and its conditional transition probabilities

$$\pi\{a_i, a_j\} = P\{X^{t+1} = a_j | X^t = a_i\} \tag{2}$$

The stationary probability of each state can be computed using the Chapman-Kolmogorov equations:

$$P\{X^{t+1} = a_j\} = \sum_i P\{X^t = a_i\}\ \pi\{a_i, a_j\} \tag{3}$$

The stationary probability for each state is obtained solving equation (3) for $X = X^{t+1} = X^t$. If the state probabilities are known, the steady state Markov chain can be modeled by an absolute transition probability, P_{ij}. This probability is obtained from the conditional transition probability and the stationary state probability using the following expression:

$$P_{ij} = P\{X = a_i\}\ \pi\{a_i, a_j\} \tag{4}$$

We will use a notation for the Markov chain where π_{ij}, representing the conditional probability, is shown near the edge representing the transition and P_{ij}, representing the absolute transition probability, is shown in parenthesis.

Assume that the primary inputs of the circuit are $X = \{x_1, \ldots, x_n\}$. For each input node i, we define two variables, one representing the value $b$efore the transition (x_i^b) and the other one representing the value $a$fter the transition (x_i^a). Each F_i, defined over the variables $X_{ba} = X_b \bigcup X_a = \{x_1^b, \ldots, x_n^b\} \bigcup \{x_1^a, \ldots, x_n^a\}$ represents all possible transitions that take place with probability P_i.

The Markov chain that models the input vectors is defined by a set $\mathcal{F}$ of pairs (F_i, P_i),

$$F_i : B^n \times B^n \rightarrow B, P_i \in [0, 1] \tag{5}$$

that define a Markov chain with

$$P\{X = X_b\}\ \pi\{X_b, X_a\} = P_{ba} = \frac{P_i}{|F_i|} \tag{6}$$

for all pairs (X_b, X_a), such that $F_i(X_b, X_a) = 1$.

This set $\mathcal{F}$ must obey the following conditions:

$$\begin{aligned} \sum P_i &= 1 \\ \forall_{X_j} \sum_{k,\, i:F_i(X_j, X_k)=1} \frac{P_i}{|F_i|} &= \sum_{k,\, i:F_i(X_k, X_j)=1} \frac{P_i}{|F_i|} \end{aligned} \tag{7}$$

By specifying a set of pairs (F_i, P_i), complex temporal correlations between input words are accurately modeled. Each F_i is specified by a description language that describe the input word before and after a transition, or by an arbitrary function described using a context free grammar.

The F_i functions are specified using either a full description of each function F_i or a simple formalism that, for each possible transition, gives a description of the input word before and after a transition. Individual bits in the word can be represented by:

- `0` : The bit takes the value 0
- `1` : The bit takes the value 1
- `-` : The bit takes any value.
- `.` : The bit keeps its value after the transition.
- `#` : The bit changes value after the transition.

Different input word transitions that occur with the same probability are modeled by a unique F_i function. To illustrate the use of this language consider the following example.

Example 1 *Consider that a circuit with* 3 *primary inputs* $\{x_3, x_2, x_1\}$ *has the following input specification.*

x_b^3	x_b^2	x_b^1	x_a^3	x_a^2	x_a^1	P_i
.	1	-	.	1	-	*0.7*
#	1	1	#	1	1	*0.3*

This description states that:

- *With probability* 0.7*, the first bit stays the same after the transition, the second bit is fixed to* 1 *and the last bit can take any value. This probability correspond to* 8 *input transitions, which means that each transition has a probability,* P_{ba}*, equal to* $0.7/8 = 0.0875$.

- *With probability* 0.3*, the first bit changes value after the transition while the last two bits stay with value* 1*. In this case there are only two input transitions with probability,* P_{ba}*, equal to* 0.15 *for each one.*

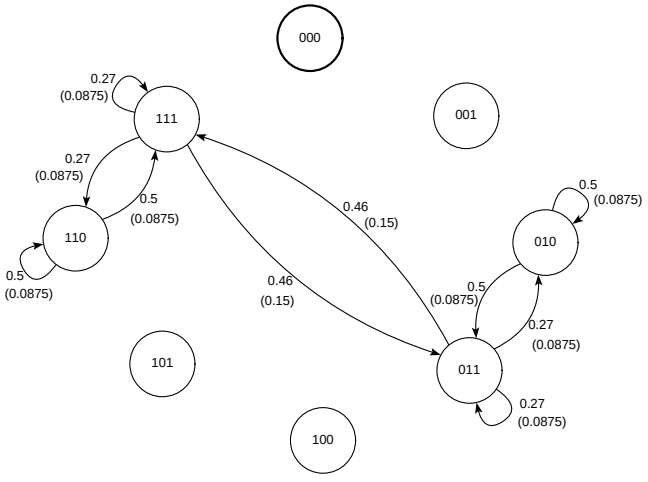

Figure 1. Markov chain for example 1.

The Markov chain described by this specification is represented in figure 1. For this circuit with 3 *primary inputs the Markov chain that specifies all possible input transitions can have up to* 8 *states. This description only specifies* 4 *possible input combinations and the correspondent transitions between them. This description is equivalent to the following one, where the functions F_i are given by:*

```
(x3b XNOR x3a) AND (x2b AND x2a)                    0.7
(x3b XOR x3a) AND (x2b AND x2a) AND (x1b AND x1a)   0.3
```

3 Formulation of the Chapman-Kolmogorov equations

Consider now a system composed of a sequential circuit whose inputs follow a distribution specified by a Markov chain as described in section 2. The behavior of a sequential circuit can be represented by its state transition graph, where the states are $Q = \{q_1, \ldots, q_M\}$.

The complete system can be modeled by a composed Markov chain that results from the composition of the state transition graph of the circuit with the Markov chain that describes the input statistics. The resulting Markov chain Y^t, is represented by a set of states $D = A \times Q = \{d_{11}, \ldots, d_{KM}\}$. The associated transition probabilities, $\pi\{d_{ij}, d_{kl}\}$ are computed using

$$\pi\{d_{ij}, d_{kl}\} = \pi\{a_i, a_k\} \tag{8}$$

where j and l are such that $q_l = \delta(q_j, a_i)$, where $\delta(q, a)$ is the transition function of the circuit. In example 2, figure 3 illustrates this computation.

The state probabilities are dependent on the structure of the Markov chain that results from the composition of the circuit STG with the Markov chain at the inputs. The stationary probability of each state can be computed by solving for $Y^{t+1} = Y^t$ the Chapman-Kolmogorov equations:

$$\begin{aligned} P\{Y^{t+1} = d_{kl}\} &= \sum_{ij} P\{Y^t = d_{ij}\}\ \pi\{d_{ij}, d_{kl}\} \\ &= \sum_{ij} P\{Y^t = d_{ij}\}\ \pi\{a_i, a_k\} \quad (9) \\ \sum_{ij} P\{Y^t = d_{ij}\} &= 1 \end{aligned}$$

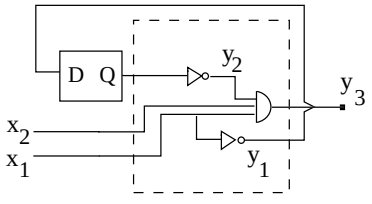

Figure 2. Example of a sequential logic circuit.

Example 2 *The STG on the top of figure 3 represents the behavior of the sequential circuit presented in figure 2. In this figure, the labels on the edges correspond to the PIs, (x_2, x_1) and the value of the output is not shown. Now, assume that at the primary inputs $\{x_2, x_1\}$ we have the following set of ICFs:*

x_2^b	x_1^b	x_2^a	x_1^a	P_i
.	#	.	#	0.5
#	.	#	.	0.5

The Markov chain on the left of figure 3 specifies this set of input correlations. The Markov chain with 8 states shown in the center of figure 3 results from the composition of the state transition graph of the circuit and the Markov chain of the input distribution. Figure 3 illustrates the composed Markov chain. The codification of each state of the composed Markov chain results from the composition of the FSM state code with the Markov state code at the inputs. For example, the state code (101) *results from the concatenation of* 1 *(FSM state code) with* 01 *(Markov state code). The transition probabilities in the composed chain are given by equation (8).*

The method we used to solve this $P = G(P)$ linear system of equations is the Picard-Peano method that finds a fixed point of the system. The method starts with an initial guess P^0, and iteratively computes

$$P^{k+1} = G(P^k) \tag{10}$$

until convergence is reached. Convergence is considered to be reached if $|P^{k+1} - P^k|$ is sufficiently small.

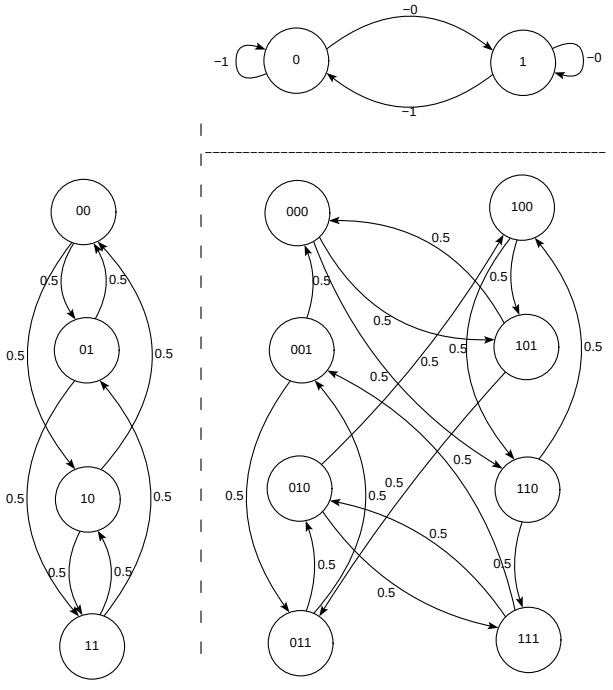

Figure 3. Markov chain of the composed system.

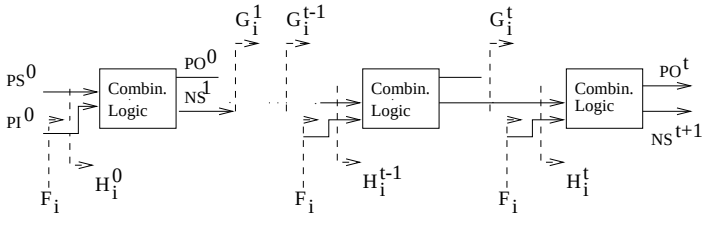

Figure 4. Unrolled sequential circuit.

3.1 Implicit resolution of the equations

The most significant contribution of this work is a method that implicitly solves the Chapman-Kolmogorov equations (9), by iteratively applying expression (10). The method is based on the fact that each iteration of expression (10) takes a probability distribution over the states of the Markov chain at time t and computes the probability distribution at time $t+1$.

We use the unrolling procedure, similar to the one proposed by other authors [8] and shown in figure 4, to model the correlations introduced by the sequential behavior of the circuit.

Consider a circuit with n input nodes, z state lines and m internal nodes. The set of variables $X = \{x_1, \ldots, x_n\}$ corresponds to the input nodes, the set of variables $S = \{s_1, \ldots, s_z\}$ corresponds to the state lines and the set of variables $Y = \{y_1, \ldots, y_m\}$ corresponds to the internal nodes. Each internal node y_i is associated with a function $f_i(X, S)$. A subset of the internal nodes are also next state lines of the circuit, $N = \{y_1, \ldots, y_z\}$.

Let $S_{ba} = S_b \bigcup S_a = \{s_1^b, \ldots, s_z^b\} \bigcup \{s_1^a, \ldots, s_z^a\}$, $Y_{ba} = Y_b \bigcup Y_a = \{y_1^b, \ldots, y_m^b\} \bigcup \{y_1^a, \ldots, y_m^a\}$ and $N_{ba} = N_b \bigcup N_a = \{y_1^b, \ldots, y_z^b\} \bigcup \{y_1^a, \ldots, y_z^a\}$.

Let $\mathcal{R}(X_b, S_b, N_b)$ be the transition relation of the circuit computed using the following expression:

$$\mathcal{R}(X_b, S_b, N_b) = \prod_{y_i^b \in N_b} y_i^b \equiv f_i(X_b, S_b) \qquad (11)$$

A set of functions G_i^t and H_i^t is used in the computation of successive iterations of equation (10). The functions G_i^t at iteration step t are defined by:

$$G_i^t = [X_b \leftarrow X_a, S_b \leftarrow S_a] \forall_{X_b, S_b} V_{ba}^t \equiv X_{ba} \mathcal{R}(X_b, S_b, S_a) H_i^{t-1} \qquad (12)$$

The functions H_i^t, that specify all possible transitions at the primary inputs for iteration step $t+1$, are defined by:

$$H_i^t = F_i G_i^t \qquad (13)$$

For the first iteration step, $H_i^0 = F_i$, and no function is imposed at the state lines, which means that all possible combinations at these lines are assumed. Subsequent iterations impose restrictions in the value of the state lines that model the sequential correlations imposed by the circuit.

Functions G_i^1 result from the first iteration step, and model the correlations imposed by the circuit in the next state lines after one clock cycle. This correlations are computed by computing the product of $\mathcal{R}(X_b, S_b, S_a)$ and $H_i^0 = F_i$, which models the input correlations. Functions G_i^1 depend on the next state variables, on the input variables X_a and on the set of the previous input variables X_{ba}. However, and in order to be able to perform the next iteration, the input variables X_a have been moved to X_b, and the next state variables S_a have been moved to the place of the current state variables, S_b. Furthermore, in order to avoid collapsing of minterms due to variable reuse, it is necessary to create a new set of variables V_b^1 and V_a^1 that is used to store the previous input variables X_{ba}. Each new iteration at time t requires a new set of variables V_b^t and V_a^t. In this way, the number of variables needed grows with the number of iterations of the fixed point procedure. However, this does not represent a significant limitation, since the algorithm usually converges after a relatively small number of iterations.

The functions H_i^1 are obtained from the product of the functions G_i^1 and the F_i functions that are presented at the primary inputs. The set of functions $\{H_1^1, H_2^1, \ldots, H_r^1\}$

represents the Markov chain that describes the input and state lines distribution for the next iteration. The probability associated with each new H_i^t is computed using:

$$P_i^t = P_i \times \frac{P_i \frac{|H_i^t|}{|F_i|}}{\mathcal{N}^t} \quad (14)$$

where $\mathcal{N}^t = \sum_{i=1}^{r} P_i \frac{|H_i^t|}{|F_i|}$ is a normalizing factor. $|H_i^t|$ denotes, in this formula, the number of minterms in H_i^t and $|F_i|$ denotes the number of minterms in F_i.

By iterating this procedure, the algorithm obtains functions H_i^t and associated P_i^t that, after convergence, represent the stationary state probabilities of the circuit.

4 Application in power estimation

The approach described in the previous section can be used to compute the power dissipation in sequential circuits considering all temporal and spatial correlations between the primary inputs, the state lines and the internal signals.

To be able to compute the node transitions we will use a function that specifies the valid combinations of variables in the circuit. This function, the Transition Consistency Function (TCF), has been proposed in [2] and depends on the input and internal signals at time t (subscripted with a b) and time $t+1$ (subscripted with an a).

For the case of a sequential circuit, the TCF is defined over the space $X_b \cup X_a \cup S_b \cup S_a \cup Y_b \cup Y_a$ and represents all possible transitions of the circuit under analysis, when the zero delay model is used. The TCF is defined to be 1 if the values of the variables in the circuit X, S, Y may experiment a transition from X_b, S_b, Y_b to X_a, S_a, Y_a. The TCF is computed using:

$$\mathcal{T}(X_{ba}, S_{ba}, Y_{ba}) = \prod_{j=a,b} \prod_{i=1}^{m} y_i^j \equiv f_i(X_j, S_j) \quad (15)$$

4.1 Computation of dissipated power

Under a simplified model, the energy dissipation of a CMOS circuit is directly related to the switching activity. The model used by most power estimation algorithms that work at the logic level is of the form:

$$P_{avg} = \frac{1}{2} \cdot (\sum_k C_k T_k) \cdot V_{DD}^2 \cdot f_{clk} \quad (16)$$

where V_{DD} is the voltage of the power supply of the module, f_{clk} is the clock frequency at which it operates, C_k the capacitance of node k and T_k the switching activity of node k.

The procedure described in the previous section is used to compute the steady state activities in each node of the circuit. For each iteration step the activity at each node of the circuit is computed in order to verify if convergence has been achieved. When convergence is reached the dissipated power is computed using expression (16).

Using the $\mathcal{T}$ function, the switching activity at node k at iteration step t, T_k^t, is given by [2]:

$$T_k^t = \sum_i |(x_k^b \oplus x_k^a)\mathcal{T}(X_{ba}, S_{ba}, Y_{ba}) H_i^t| \times \frac{P_i^t}{|H_i^t|} \quad (17)$$

where $x_k \in \{X \cup S \cup Y\}$.

After the computation of the product of functions $\mathcal{T}(X_{ba}, S_{ba}, Y_{ba})$ and H_i, the number of transitions for each node is counted. The result is multiplied by the probability P_i of function H_i and normalized.

A generalization of the above method to handle non-zero delay models has been proposed in [1] and is based on a mechanism that applies discretization to the time axis.

5 Results

This section presents preliminary results obtained using the approach described in section 3 and the power estimates obtained as described in section 4. To evaluate the applicability of the method the algorithm was integrated within the SIS logic synthesis system [7]. As test cases, we used a set of circuits from the ISCAS'89 benchmark. These circuits were mapped to the MSU library after application of the script *script.algebraic*. All tests were performed on a 300MHz Pentium running Linux with a CPU time limit of 4 hours and memory usage restricted to 90 MB.

Table 1 summarizes the obtained results. Columns 2 to 5 in this table list the statistics of the circuits, namely the number of primary inputs, primary outputs, registers and size, in literals. The next six columns list the number of iterations and number of BDD variables required to solve the Chapman-Kolmogorov equations for the test circuits when three different sets of $ICFs$ are present at the circuit inputs. Finally, the last six columns list the power estimates obtained (in μW) and the CPU times used (in seconds).

The $ICFs$ specified describe respectively, a One Hot code (where only one bit at a time takes the value one), a code called High/Low (where two input bits exhibit high activity while the remaining input bits are stationary[1]) and a Binary code. These specifications correspond to Markov chains with a number of states equal to n, 2^n and 2^n, respectively, where n is the number of inputs.

From the results in this table it is possible to verify that even when we the system is described by Markov chains with a large number of states the proposed method is able to implicitly solve the Chapman-Kolmogorov system of equations, converging to the steady state probabilities in a small

[1]This code models a situation that is very common and very hard to model using either simulation based methods or random sampling.

Circuit	# I	# O	# Reg	# Lits	Onehot		High/Low		Bin		Onehot		High/Low		Bin	
					Iter	Vars	Iter	Vars	Iter	Vars	P	cpu	P	cpu	P	cpu
shiftreg	1	2	3	8	5	28	3	24	5	28	0	2	58	2	40	2
train4	2	1	2	22	3	64	14	108	3	64	35	2	92	24	75	2
bbtas	2	2	3	33	5	96	30	196	-	-	33	7	106	344	-	-
train11	2	1	4	68	5	168	21	232	9	184	190	22	168	370	242	48
beecnt	4	28	3	71	5	190	8	214	4	182	271	27	189	82	315	25
dally	14	6	4	73	10	462	7	378	7	378	180	738	41	5598	63	5732
ex5	2	4	4	93	-	-	11	242	10	238	-	-	278	272	470	144
dk17	2	3	3	101	-	-	10	252	10	252	-	-	356	226	422	116
s208	10	1	8	117	3	330	3	330	3	330	55	440	119	1101	131	1985
s298	3	6	14	184	6	438	-	-	8	450	431	2487	-	-	269	6517
s420	18	1	16	233	3	642	-	-	-	-	49	13327	-	-	-	-
cse	7	7	4	258	4	594	6	622	20	818	229	492	14	3166	123	14305
s382	3	6	21	306	5	690	-	-	4	684	390	12198	-	-	397	5466
kirkman	12	6	4	311	14	99	-	-	-	-	475	9814	-	-	-	-
keyb	7	2	5	316	16	880	-	-	10	796	807	8934	-	-	489	9525
ex1	8	19	5	319	11	840	-	-	-	-	431	3026	-	-	-	-
s641	35	23	19	539	4	1466	-	-	-	-	51	13076	-	-	-	-

Table 1. Statistics of the circuits, number of iterations and variables in the BDDs, and results obtained.

number of iterations. To illustrate this point, consider the circuit *s382* with Binary code at the inputs. This circuit is described by a Markov chain with 2^{24} states and only 4 iterations were required to find a solution. The number of BDD variables grows for each new iteration due to the need to store the previous input variables to avoid collapsing of minterms due to variable reuse.

The results also show that if the correlations between the input sequences and the states are not considered a significant error is introduced in the power estimates, because codes with similar activities (e.g., OneHot and Binary) give very different results.

6 Conclusions and future work

We have presented an algorithm that implicitly solves the Chapman-Kolmogorov equations to compute the stationary state probability distribution of a system composed by a sequential circuit whose input sequence is modeled by a discrete Markov chain. By composing the input Markov chain with the state transition graph that describes the sequential behavior of the circuit, it is possible to model exactly all the temporal and spatial correlations between the signals.

The application of the new approach to power estimation results in a method that performs power estimation for sequential circuits using input statistics that would require extremely large traces if simulation based methods were used.

An interesting direction for future work resides on the use of approximation algorithms to achieve a larger range of applicability of this method. Another interesting direction is the application of this method to the analysis of networks of finite state machines. In fact, the method described in section 3 can be generalized to solve the equations that result from the composition of the Markov chains that model the behavior of any number of interacting finite state machines.

References

[1] A. T. Freitas. *Estimação de potência em circuitos digitais utilizando técnicas de enumeração implícita*. PhD thesis, IST, Lisbon Technical University, Nov 2002.

[2] A. T. Freitas and A. L. Oliveira. Circuit partitioning techniques for power estimation using the full set of input correlations. In *IEEE International Conference on Electronics, Circuits, and Systems*, volume II, pages 903–908, Sept. 2001.

[3] A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of average switching activity in combinational and sequential circuits. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 253–259, June 1992.

[4] G. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Markovian analysis of large finite state machines. *IEEE Transactions on Computer-Aided Design*, 15(12):1479–1493, Dec. 1996.

[5] D. Marculescu, R. Marculescu, and M. Pedram. Trace-driven steady-state probability estimation in fsms with application to power estimation. In *Proceedings of the Design, Automation and Test in Europe*, pages 774–781, Paris, 1998.

[6] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2^{nd} edition, 1984.

[7] E. Sentovich, K. Singh, C. Moon, H. Savoj, R. Brayton, and A. Sangiovanni-Vincentelli. Sequential Circuit Design Using Synthesis and Optimization. In *Proceedings of the International Conference on Computer Design: VLSI in Computers and Processors*, pages 328–333, October 1992.

[8] C.-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. Despain, and B. Lin. Power estimation methods for sequential logic circuits. *IEEE Transactions on VLSI Systems*, 3(3):404–416, Sept. 1995.