# Trace-Driven Steady-State Probability Estimation in FSMs with Application to Power Estimation*

Diana Marculescu, Radu Marculescu, Massoud Pedram

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

**Abstract** - *This paper illustrates, analytically and quantitatively, the effect of high-order temporal correlations on steady-state and transition probabilities in finite state machines (FSMs). As the main theoretical contribution, we extend the previous work done on steady-state probability calculation in FSMs to account for complex spatiotemporal correlations which are present at the primary inputs when the target machine models real hardware and receives data from real applications. More precisely: 1) using the concept of constrained reachability analysis, the correct set of Chapman-Kolmogorov equations is constructed; and 2) based on stochastic complementation and iterative aggregation/ disaggregation techniques, exact and approximate methods for finding the state occupancy probabilities in the target machine are presented. From a practical point of view, we show that assuming temporal independence or even using first-order temporal models is not sufficient due to the inaccuracies induced in steady-state and transition probability calculations. Experimental results show that, if the order of the source is underestimated, not only the set of reachable sets is incorrectly determined, but also the steady-state probability values can be more than 100% off from the correct ones. This strongly impacts the accuracy of the total power estimates that can be obtained via probabilistic approaches.*

## 1. Introduction

In the last decade, probabilistic approaches have received a lot of attention as a viable alternative to deterministic techniques for analyzing complex digital systems. Logic synthesis [1], verification [2], testing [3] and more recently, low-power design [4] have benefited from using probabilistic techniques. In particular, the behavior of FSMs has been investigated using concepts from the Markov chain (MC) theory.

Studying the behavior of the MC provides us with different variables of interest for the original FSM. In this direction, [5][6] are excellent references where steady-state and transition probabilities (as variables of interest) are estimated for large FSMs. Both techniques are analytical in nature but, in order to manage complexity, make some simplifying assumptions, temporal independence of the primary inputs being the most notable one. As we will show in this paper, temporal correlations longer than one time step can significantly affect the overall behavior of the FSM and therefore result in very different values for the actual transition probabilities compared to those predicted in [5][6]. More interestingly, it will be shown that if one ignores the effect of finite-order statistics at the primary inputs of the FSM, it is possible to wrongly predict non-zero steady-state probabilities for some transient states (which normally occur in the beginning of operation, but disappear once the machine reaches its steady-state regime). Knowledge of correct state occupancy probabilities is important in timing verification, state assignment, re-encoding for low-power and power estimation via probabilistic or statistical approaches.

Addressing these issues, the present paper extends the previous work reported in [5] to explicitly incorporate complex spatiotemporal correlations in steady-state and transition probabilities calculation for FSMs. The analysis itself relies on time-homogeneous discrete-parameter MCs which are used in two different ways:

1) an input modeling MC is used to model the binary input stream that typifies the application data (called also *trace*) feeding the target FSM[1];

2) a composite MC is used to model the serial connection of the input-modeling MC and the MC of the FSM itself.

Studying the composite MC requires reachability analysis on the target machine. At this point, our work differs substantially from what other researchers have considered in the past, in the sense that our *reachability analysis* is *constrained* by the actual input sequence and accounts for the very specific way in which the input source excites the target FSM.

Last but not least, analysis of MCs involves sophisticated numerical techniques. To date, Gauss-Jacobi and power method have been extensively used in steady-state probability calculation [5]. We present instead two different algorithms based on *stochastic complementation* and *iterative aggregation/ disaggregation*, which provide a deep insight into theoretical aspects of MC analysis and an efficient solution for a large class of MCs, i.e. *nearly completely decomposable* (NCD) systems. The present paper thus improves the state-of-the-art in two ways:

1) using the concept of *constrained reachability analysis*, it constructs the correct set of Chapman-Kolmogorov equations;

2) based on *stochastic complementation* and *iterative aggregation/disaggregation* techniques, it presents *exact* and *approximate* techniques for finding the state occupancy probabilities in the target machine.

The paper is organized as follows. Section 2 presents the basic definitions and notations on FSMs and MCs that will be used throughout the paper. In Section 3 we formulate the problem we want to solve and present the basic Markov model. Section 4 focuses on constrained reachability analysis issue. In Section 5 we present exact and approximate methods for steady-state probabilities calculation, and we point out some issues regarding complexity and convergence of algorithms. Finally, we present some experimental results for common sequential benchmarks, and we conclude by summarizing our main contribution.

## 2. Preliminaries on finite-order MCs

In this section we present the basic definitions and notation. For a complete documentation, we refer the reader to [7][8]. We consider only time-homogeneous discrete-parameter MCs with finite state space and assume that all states are *recurrent* (i.e. the probability of returning to it after $n \geq 1$ steps is greater than zero) since all *transient* states vanish after a finite number of steps.

*This research was supported by DARPA under contract no. F33615-95-C1627 and NSF under award no. MIP-9628999.

[1]We point out that although the Markov model is derived for a particular input trace, it is completely general and represents in a compact form the whole class of input sequences having the same characteristics.

***Definition 1***. A discrete stochastic process $\{x_n\}_{n \geq 0}$ is said to be a lag-$k$ MC if at any time step $n \geq k$:

$$p(x_n = \alpha_n | x_{n-1} = \alpha_{n-1} x_{n-2} = \alpha_{n-2} \ldots x_0 = \alpha_0) = p(x_n = \alpha_n | x_{n-1} = \alpha_{n-1} x_{n-2} = \alpha_{n-2} \ldots x_{n-k} = \alpha_k) \quad (1)$$

If $k = 1$, the conditional probabilities $p(x_n = \alpha_n | x_{n-1} = \alpha_{n-1})$ are called *single-step transition probabilities* and represent the conditional probabilities of making a transition from state $x_{n-1}$ to state $x_n$ at time step $n$. In homogeneous MCs these probabilities are independent of $n$ and consequently written as $p_{ij} = p(x_n = j | x_{n-1} = i)$ for all $n$ = 1, 2,... The matrix $Q = \{p_{ij}\}$ is called the *transition probability matrix*. We note that $Q$ is a *stochastic matrix* because its elements satisfy the following two properties: $0 \leq p_{ij} \leq 1$ and $\sum_j p_{ij} = 1$.

An equivalent description of the MC can be given in terms of its *state transition graph* (STG). Each node in the STG represents a state in the MC, and each edge from node $i$ to node $j$ is labelled with the one-step transition probability from state $i$ to state $j$ ($p_{ij}$). It should be noted that any lag-$k$ MC can be reduced to a lag-one MC based on the following result.

***Proposition 1.*** [8] If $\{u_n\}_{n \geq 1}$ is a lag-$k$ MC then $\{v_n\}_{n \geq 1}$, where $v_n = (u_n, u_{n+1}, ..., u_{n+k-1})$, is a multivariate lag-one MC.

For clarity, we will thus refer subsequently only to lag-one MCs; using Proposition 1, all results translate to lag-$k$ MCs.

***Definition 2.*** A MC is said to be *nearly completely decomposable* if its transition matrix $Q$ can be written in the form

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & \ldots & Q_{1p} \\ Q_{21} & Q_{22} & \ldots & Q_{2p} \\ \ldots & \ldots & \ldots & \ldots \\ Q_{p1} & Q_{p2} & \ldots & Q_{pp} \end{bmatrix}$$

, where all non-zero elements in the off-diagonal blocks are small (the precise meaning of 'small' will be defined later in Section 5) compared to those in the diagonal blocks. If the off-diagonal blocks are exactly zero, then the MC is said to be *completely decomposable*.

We now turn our attention to distribution defined on the states of a MC. We shall denote by $\pi_i(n)$ the probability that the MC is in state $i$ at step $n$, that is $\pi_i(n) = p(x_n = i)$. In vector notation, $\pi(n) = (\pi_1(n), \pi_2(n), ..., \pi_i(n), ...)$, where $\pi$ is a row vector and $\pi(n) = \pi(0) \cdot Q^n$, where $\pi(0)$ denotes the initial state distribution of the chain. For nondecomposable and aperiodic MCs, it is shown that the *limiting distribution* $\pi = \lim_{n \to \infty} \pi(n)$ always exists and it is independent of the initial probability distribution [7]. In addition, the following important result holds.

***Proposition 2.*** [7] For a nondecomposable MC, the equation $\pi \cdot Q = \pi$ with $\sum_j \pi_j = 1$ has a unique solution that represents the *stationary distribution* of the MC.

The unique solution of the equation in Proposition 2 (called the set of Chapman-Kolmogorov equations) can actually be determined by solving the system of equations $\pi_j = \sum_i \pi_i \cdot p_{ij}$ with $\sum_j \pi_j = 1$.

## 3. FSM steady-state analysis: problem formulation

In this section, we introduce formally the problem we want to solve and present two Markov models that we use to solve it: one associated to the state lines of the FSM and another one for the input sequence that feeds the target FSM.

The probabilistic behavior of an FSM can be studied by regarding its STG as a MC. More precisely, by attaching to each out-going edge of each state in the target FSM a transition probability that corresponds to that particular transition, one can actually obtain a MC as defined in previous section. Furthermore, studying the behavior of the underlying MC gives us different variables of interest for the original FSM. To this end, the set of equations to solve is the one from Proposition 2 (where $Q = Q_S$ is the transition matrix associated to the FSM).

To set up the matrix $Q_S$, the authors in [5] consider that all external input combinations are equiprobable during the normal operation of the machine and therefore, the one-step transition probability matrix can be obtained from the transition relation in a straightforward manner. However, in practice, the situation can be quite different: various input sequences may exercise the machine in different ways and thus produce substantially different STG structures. Due to the feedback lines, the behavior of the state lines themselves is strongly dependent on the characteristic of the input sequences present at primary inputs and therefore, to set up the $Q_S$ matrix which actually accounts for the influence of correlations at the primary inputs on the state lines of the FSM is a key (and nontrivial!) task. To construct the correct $Q_S$ matrix which accounts for the effect of the actual input trace, we have to associate a finite-order MC to the primary input stream.

## 4. Sequence-driven reachability analysis

In this section we first present a theoretical framework for sequence-driven reachability analysis, followed by a practical solution to this problem. We point out that sequence-driven reachability analysis differs from classical reachability analysis in that it accounts for constraints on the inputs, that is, the possible set of input vectors applicable to the circuit and their sequencing.

In [9] it has been shown that to any first-order MC, one can associate a *Stochastic Sequential Machine* (SSM) that generates symbols according to the conditional probabilities of the initial MC. Specifically, a synthesis procedure for the SSM modeling the input sequence has been proposed by the authors. Based on this, the target FSM and its input can be viewed as in Fig.1a.

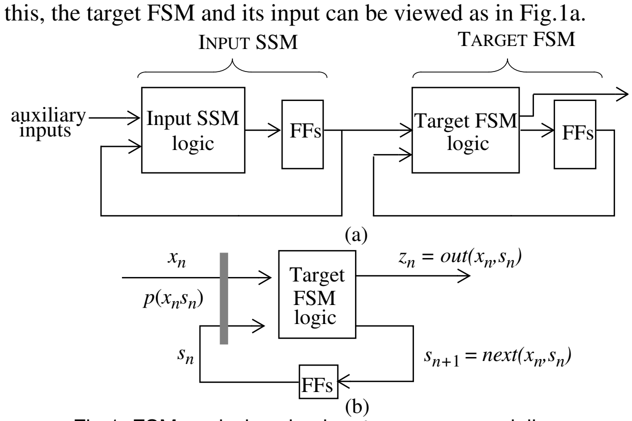



Fig.1: FSM analysis using input sequence modeling

The primary inputs of the SSM in Fig.1a (called *auxiliary inputs*) are generated according to a prescribed probability distribution such that the states of the SSM have exactly the desired probability distribution. Thus, the analysis can be done on the *product machine* (*input SSM*, *target FSM*) which has temporally uncorrelated inputs with a prescribed probability distribution. What we are interested in is the probability distribution on the state lines of the target FSM.

Referring to the general FSM in Fig.1b, we proposed until now two interacting Markov models: one for the primary inputs $\{x_n\}_{n \geq 0}$ and another one for the states $\{s_n\}_{n \geq 0}$ (which characterizes the behavior of the machine itself). In fact, these two models can be conceptually merged via joint transition probabilities $p(x_n s_n)$ and $p(x_n s_n | x_{n-1} s_{n-1})$. These probabilities completely characterize the input that feeds the next state and the output logic of the target circuit.

For convenience, we introduce a new formulation based on matrices. As we can see in Fig.1b, the first-order MC at the primary inputs $x_n$ can be characterized by the matrix of conditional probabilities $Q_X = (q_{ij})_{1 \leq i,j \leq l}$, where $q_{ij} = p(x_n = v_j | x_{n-1} = v_i)$ and $\{v_1, v_2, ..., v_l\}$ represents the set of possible input patterns. On the other hand, the MC defined jointly for the primary inputs and state lines $\{x_n s_n\}_{n \geq 0}$ can be characterized by the matrix $Q_{XS} = (r_{ip,jq})_{\substack{1 \leq i,j \leq l \\ 1 \leq p,q \leq m}}$, where $r_{ip,jq} = p(x_n s_n = v_j u_q | x_{n-1} s_{n-1} = v_i u_p)$ and $\{u_1, u_2, ..., u_m\}$ is the set of reachable states ($Q_{XS}$ is the stochastic matrix of the product machine mentioned above). From this joint characterization, we can easily derive the state probabilities as: $p(s_n) = \sum_{all\ x_n} p(x_n s_n)$ which are actually our variables of interest. Based on Theorem 2 in [17], the following result provides the starting point in finding the correct matrix $Q_{XS}$.

***Proposition 3***. The matrix $Q_{XS}$ can be written in the form

$$Q_{XS} = \begin{bmatrix} q_{11}B_1 & q_{12}B_1 & \dots & q_{1l}B_1 \\ q_{21}B_2 & q_{22}B_2 & \dots & q_{2l}B_2 \\ \dots & \dots & & \dots \\ q_{l1}B_l & q_{l2}B_l & \dots & q_{ll}B_l \end{bmatrix}$$

, where $\{B_i\}_{1 \leq i \leq l}$ is a set of $m \times m$ degenerate (i.e., containing only 0s and 1s) stochastic matrices defining the next state function for input $v_i$. Specifically, if $B_i = (b^i_{pq})_{1 \leq p,q \leq m}$ then

$$b^i_{pq} = \begin{cases} 1 & \text{if } next(v_i, u_p) = u_q \\ 0 & \text{otherwise} \end{cases}.$$

***Corollary 1.*** The product machine (*input SSM, target FSM*) (as in Fig.1a) is also a SSM whose auxiliary inputs are excited using the same probability distribution as the one used for the input SSM.

From this point on, to compute the steady-state probability for the state lines of the target machine, we can apply any existing approach that computes the probabilities for the states of the product machine (*input SSM, target FSM*) which has the virtue of having temporally uncorrelated inputs. However, this approach can be very inefficient: the task of synthesizing the exact input SSM may require huge memory and computation time. Instead, we propose to model the input as a *Dynamic Markov Tree* of order 1 ($DMT_1$) [10]. The $DMT_1$ model contains information about not only the possible binary vectors that can appear on the inputs of the FSM, but also the sequencing of these vectors. Additionally, the wordwise conditional probabilities for the primary inputs are easily extracted from such a model. The benefits of using $DMT_1$ for input modeling are threefold:

- the structure $DMT_1$ is constructed "on demand", therefore it offers a very compact representation;
- the model provides a set of parameters that completely capture spatiotemporal correlations;
- its structure is compatible with that of *Binary Decision Diagrams* (BDDs) [11] which have been successfully used in reachability analysis for FSMs [12][13].

To see how these advantages can be exploited, consider the following example.

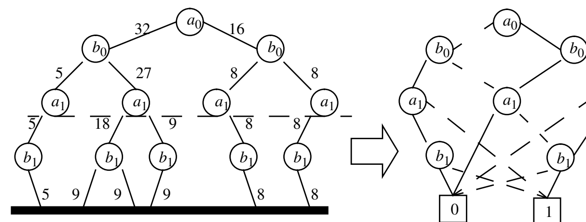
_Example 1:_ For the sequence $S_1$, the $DMT_1$ is given in Fig.2a. The corresponding BDD for this $DMT_1$ is depicted in Fig.2b. Every possible combination with non-zero probability of occurrence in $DMT_1$ is part of the ON-set of the corresponding BDD; everything else, represents the OFF-set.

$S_1$

$v_{48}$ ... $v_3 v_2 v_1$

011011011000011000000110110000000000011000000000110

1101101101111101101011011010111010111010110101101




(a) (b)

Fig.2: The tree $DMT_1$ and the corresponding BDD

The BDD corresponding to a given $DMT_1$ actually represents the transition relation $\delta$ of the machine that models the *input* to the target FSM ("INPUT SSM" in Fig.1a). Having this representation for the input, our task is then to find all the reachable combinations (*input, state*). Let $B = \{0, 1\}$, $N$ the number of primary input variables and $\delta: B^N \times B^N \rightarrow B$ defined as $\delta(x^-, x^+) = 1$ if vector $x^+$ can follow $x^-$ on the input modeled as a lag-one MC (that is, if the corresponding conditional probability is non-zero) and zero otherwise. Also, let $M$ be the number of state variables and *next*: $B^M \times B^N \rightarrow B^M$ be the next state function of the FSM. Then, we may employ the following standard procedure [13] to compute the set $C$ of reachable combinations (*input, state*) for the given FSM and input characterization:

$C_0 = \{(x_0, s_0) \mid x_0$ is any possible initial input, $s_0$ is any possible initial state$\}$,

$C_{i+1} = C_i \cup \{(x, s) \mid \exists (x', s') \in C_i$ s.t. $\delta(x', x) = 1$ and $next(s', x') = s\}$,

$C = C_i$ if $C_i = C_{i+1}$.

The above steps can be performed completely symbolically with the aid of BDDs [12][13]. After having the complete set $C$ of possible (*input, state*) combinations, we can easily build the matrix $Q_{XS}$ based on Proposition 3. We should point out that in general, the matrix $Q_{XS}$ 1) may have transient states or 2) may be decomposable. However, these can be dealt with in a similar way to the approach presented in [5] where: 1) transient states are eliminated and 2) the problem is reduced to finding the steady state distribution for each strongly connected component of the underlying MC. In what follows, we will thus refer only to irreducible MCs or matrices, that is, those in which each state is reachable from any other state in a finite number of steps. This hypothesis has no theoretical limitation to be extended to the case of reducible MCs or MCs with multiple components.

## 5. Steady-state probability computation

We have constructed by now the correct matrix $Q_{XS}$ (which completely characterizes the FSM behavior) and are therefore ready to solve the basic equation $\pi \cdot Q_{XS} = \pi$ (with $\Sigma\pi_i = 1$).

### 5.1 Classical methods

Finding the stationary distribution of MCs with relatively few states is not a difficult problem and standard techniques based on solving systems of linear equations do exist [15]. To find the stationary distribution of a MC, one can always employ direct or iterative methods to solve the Chapman-Kolmogorov equations. Both types of methods work with the stochastic matrix as a whole. However, when the matrix size is large, we must resort to decompositional methods that try to solve smaller problems and then aggregate their solutions to find the needed stationary distribution.

### 5.2 Stochastic complementation

For large-scale problems, it is natural to attempt to somehow uncouple (or decompose) the original MC into smaller chains (which are therefore easier to analyze) and finally, having these partial solutions, to produce the global stationary distribution that corresponds to the original chain.

***Definition 3***. [16] Let $Q$ be a $n \times n$ irreducible stochastic matrix partitioned as

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & \dots & Q_{1p} \\ Q_{21} & Q_{22} & \dots & Q_{2p} \\ & \dots & & \dots \\ Q_{p1} & Q_{p2} & \dots & Q_{pp} \end{bmatrix} \qquad (2)$$

where all diagonal blocks are square. For a given index $i$, let $Q_i$ denote the principal block submatrix of $Q$ obtained by deleting the $i^{th}$ row and $i^{th}$ column of blocks from $Q$, and let $Q_{i*}$ and $Q_{*i}$ designate:

$$Q_{i*} = [Q_{i1} Q_{i2} \dots Q_{i,i-1} Q_{i,i+1} \dots Q_{ip}] \text{ and} \qquad (3)$$

$$Q_{*i} = \left[Q_{1i}\ Q_{2i} \dots\ Q_{i-1,i}\ Q_{i+1,i} \dots\ Q_{pi}\right]^T$$

That is, $Q_{i*}$ is the $i^{th}$ row of blocks with $Q_{ii}$ removed and $Q_{*i}$ is the $i^{th}$ column of blocks with $Q_{ii}$ removed. The stochastic complement of $Q_{ii}$ is defined to be the matrix

$$S_{ii} = Q_{ii} + Q_{i*} \cdot (I - Q_i)^{-1} \cdot Q_{*i} \qquad (4)$$

where $I$ is the unit matrix.

It can be shown that every stochastic complement in $Q$ is also an irreducible matrix. In addition, the following theorem has been proven [16].

***Theorem 1.*** Let $Q$ be an $n \times n$ irreducible stochastic matrix as in (2) whose stationary probability vector $\pi$ can be written as $\pi = (\xi_1\Phi_1, \xi_2\Phi_2, \dots, \xi_p\Phi_p)$ with $\Phi_i \cdot e = 1$ for $i = 1, 2, \dots, p$; $e$ is a column vector defined as: $e = (1,1,\dots,1)^T$. Then $\Phi_i$ is the unique stationary probability vector for the stochastic complement $S_{ii}$ and $\xi = (\xi_1, \xi_2, \dots, \xi_p)$ is the unique stationary probability vector for the $p \times p$ irreducible stochastic matrix $A$ (called the *coupling matrix*) whose entries are defined by $a_{ij} = \Phi_i \cdot Q_{ij} \cdot e$.

The coupling matrix $A$ corresponds to an MC in which states belonging to the same block of the partition are *collapsed* into a single state. Thus, $\xi$ describes the steady-state probability of being in such a set of states. Using this important theorem, the following exact procedure can be used to compute the stationary probability vector. (The input to this procedure is the matrix $Q$ given in Definition 3).

1. Partition $Q$ into a $p \times p$ block matrix with square diagonal blocks.
2. Form the stochastic complement of each diagonal block:
   $S_{ii} = Q_{ii} + Q_{i*} (I-Q_i)^{-1} Q_{*i}$, $i = 1, 2, \dots, p$.
3. Compute the stationary probability vector of each stochastic complement:
   $\Phi_i \cdot S_{ii} = \Phi_i$; $\Phi_i \cdot e = 1$, $i = 1, 2, \dots, p$.
4. Form the coupling matrix $A$ whose elements are given by:
   $a_{ij} = \Phi_i \cdot Q_{ij} \cdot e$.
5. Compute the stationary probability vector of $A$:
   $\xi \cdot A = \xi$; $\xi \cdot e = 1$.
6. Construct the stationary probability vector of $Q$ as:
   $\pi = (\xi_1\Phi_1, \xi_2\Phi_2, \dots, \xi_p\Phi_p)$.

Fig.3: The stochastic complementation algorithm

We point out that the analysis based on stochastic complementation does not depend in any way on matrix $Q$ being NCD and when implementing the stochastic complement approach, we may choose a partitioning solution that is convenient for us. For instance, based on the functionality of the FSM, we may partition matrix $Q_{XS}$ such that combinations $(x_i, s_i)$, $(x_j, s_j)$ are put in the same block if and only if $s_i = s_j$. In this case, the matrix $Q_{XS}$ is partitioned as in (2) where each submatrix $Q_{ij}$ ($1 \le p, q \le m$) has the form (using the notations in Proposition 3)[1]:

$$Q_{ij} = \begin{bmatrix} b_{ij}^1 \cdot q_{11} & b_{ij}^1 \cdot q_{12} & \dots & b_{ij}^1 \cdot q_{1l} \\ b_{ij}^2 \cdot q_{21} & b_{ij}^2 \cdot q_{22} & \dots & b_{ij}^2 \cdot q_{2l} \\ \dots & \dots & \dots & \dots \\ b_{ij}^l \cdot q_{l1} & b_{ij}^l \cdot q_{l2} & \dots & b_{ij}^l \cdot q_{ll} \end{bmatrix} \qquad (5)$$

The following remarkable result holds:

***Theorem 2.*** If the stochastic complementation algorithm is applied to matrix $Q_{XS}$ and the partitioning is done as in (5), then $A$ is the matrix associated with the MC for the state lines $Q_S$ and the corresponding probability distribution is given by $\xi$.

This important result justifies basically the applicability of stochastic complementation to FSM analysis. We also note that this method has the important feature of being *exact,* but unfortunately, it is computationally inefficient on monoprocessor machines (it involves the inversion of a large matrix $(I - Q_i)$). Its contribution lies primarily in the insight it provides into theoretical aspects of NCD systems.

### 5.3 Iterative aggregation/disaggregation

In this section, we present an iterative algorithm based on approximate decomposition that rapidly converges to the exact solution when the MC is NCD. The pioneering work on NCD systems comes from Simon and Ando [18] who studied the dynamic behavior of linear systems. The idea behind the dynamic behavior of NCD systems is the existence of two operational regimes:

- a *short-run dynamics,* when strong interactions within each subsystem are dominant and quickly force each subsystem to a local equilibrium almost independently of what is happening in the other subsystems;

- a *long-run dynamics*, when weak interactions among groups

[1]This partitioning is *state-oriented*, while the one in Proposition 3 is *input-oriented.*

begin to become important and the whole system moves toward a global equilibrium; in this global equilibrium the relative values attained by the states at the end of the short-run dynamics period are maintained.

As a consequence, the state space of the global MC can be partitioned into disjoint sets, with strong interactions among the states of a subset, but with weak interactions among the subsets themselves. This way, each subproblem can be solved separately and the global solution is then constructed from partial solutions.

Iterative aggregation/disaggregation (IAD) methods are particularly useful when the global MC is NCD. More precisely, IAD methods work on partitioned state space as an *aggregation* (or *coupling*) step followed by a *disaggregation* one. The coupling step involves generating a stochastic matrix of block transition probabilities (the *coupling* or *aggregation matrix*) and then determining its stationary probability vector. The disaggregation step computes an approximation to the probability of each state aggregated within the same block. The basic iterative algorithm is called KMS (after its authors Khoury-McAllister-Stewart) [14] and is described in Fig.4 (Once again, the input of the algorithm is the matrix $Q$ as in Definition 3.)

1. Let $\pi^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)},...,\pi_p^{(0)})$ be a given initial approximation to the solution $\pi$, and set $m = 1$.
2. Compute $\Phi^{(m-1)}$: for $i = 1, 2,..., p$ do

$$\Phi_i^{(m-1)} = \pi_i^{(m-1)} / \left\|\pi_i^{(m-1)}\right\|_1$$

(the norm is defined as: $\|x\|_1 = \sum_j |x_j|$ ).

3. Construct the aggregation matrix $A^{(m-1)}$ whose elements are given by:

$$(A^{(m-1)})_{ij} = \Phi_i^{(m-1)} \cdot Q_{ij} \cdot e.$$

4. Solve the eigenvector problem:

$$\xi^{(m-1)} \cdot A^{(m-1)} = \xi^{(m-1)}; \quad \xi^{(m-1)} \cdot e = 1.$$

5.(a) Compute the row vector:

$$z^{(m)} = (\xi_1^{(m-1)}\Phi_1^{(m-1)}, \xi_2^{(m-1)}\Phi_2^{(m-1)}, \ldots, \xi_p^{(m-1)}\Phi_p^{(m-1)}).$$

(b) Find $\pi^{(m)}$: for $k = 1, 2,..., p$ solve the system of equations:

$$\pi_k^{(m)} = \pi_k^{(m)} \cdot Q_{kk} + \sum_{j<k} \pi_j^{(m)} \cdot Q_{jk} + \sum_{j>k} z_j^{(m)} \cdot Q_{jk}$$

6. Test for convergence: if the accuracy is sufficient, then stop and take $\pi^{(m)}$ as the solution vector; otherwise, set $m = m + 1$ and go to step 2.

Fig.4: The KMS algorithm

In this case, the partitioning criterion is purely numerical. States are aggregated within the same block if they interact strongly enough to favor short-run dynamics. This way, the off-diagonal elements in the global matrix are smaller than those in the blocks on the main diagonal and therefore, the interactions among subsets are minimized. In practice, finding the partitioning of a NCD stochastic matrix is not a trivial task. One way to do it is to ignore the entries in the matrix that are less than some threshold $\varepsilon$ and then find the strongly connected components of the underlying MC [15]. Next, the threshold may be increased and the same analysis is done on the components already found. We should note that the lower the threshold, the higher the rate of convergence, at the expense of larger blocks. On the other hand, if the partitioning is done until the blocks become manageable, the convergence rate slows down due to the larger threshold used. More formally, under fairly general conditions, the following result holds for NCD matrices:

***Theorem 3.*** [14] If the matrix $Q$ is partitioned such that $\|Q_{ii}\|_1 = O(1)$ and $\|Q_{ij}\|_1 = O(\varepsilon)$ for $i \neq j$, then the error in the approximate solution using the iterative aggregation/disaggregation algorithm is reduced by a factor of $\varepsilon$ at each iteration.

In practice, the KMS algorithm offers the attractive feature of working on individual blocks instead of the whole matrix $Q_{XS}$. As a consequence, its complexity per iteration is given by $O(n_{max}^3)$, where $n_{max}$ is the maximum order over all diagonal blocks in the partitioned matrix $Q_{XS}$. Compared to the classical power method (which is also iterative in nature) [5], the KMS algorithm has the significant advantage of being applicable to *any irreducible* MC (aperiodic or periodic) and also having a *higher rate of convergence* for NCD systems. In these cases, a few iterations will suffice for all practical purposes. We should point out that we can always trade-off space versus time complexity: if the partitioning is such that $n_{max}$ is still too large, we can use a higher value for the threshold such that the size of the largest subset (i.e. $n_{max}$) becomes manageable. In this case, the convergence rate will be smaller and thus, the time needed for convergence will increase. Also, the size of the coupling matrix will be larger and hence step 4 in the KMS algorithm becomes critical. However, there is a solution to this problem: when solving the Chapman-Kolmogorov equations for the coupling matrix, we can apply the KMS algorithm in a recursive manner. This approach is called the *hierarchical KMS algorithm* [15].

## 6. Experimental results

In this section we provide our experimental results for some common benchmark circuits. In particular, we compare the probability distribution for the states where the order of the input sequence is assumed by default to be one, against the case where the actual order of the source is taken into consideration.

We provide in Table 1 the results obtained for stochastic complementation when Fibonacci-type sequences (that is, second-order sequences) are applied at the primary inputs of some *mcnc*'91 and ISCAS'89 benchmarks. We considered for this only small benchmarks because the exact method based on stochastic complementation is, in general, computationally inefficient. For each example, an appropriate dynamic Markov tree has been built and based on it, a sequence-driven reachability analysis has been performed. Using the obtained set of reachable combinations (*input, state*) (denoted by #(*x, s*)), and sparse matrix techniques, the matrix $Q_{XS}$ has been built and further used to determine steady-state probabilities. The partitioning was determined by the same functional criterion as in Theorem 2. We also report for each benchmark the number of reached states (#*s*) and their corresponding probabilities.

Table 1: Steady-state distributions obtained with stochastic complementation for second-order input sequences

| Circuit | PIs/ FFs | #(*x*,*s*) | #*s* | State probability distribution |
|---|---|---|---|---|
| bbara | 4/4 | 28 | 3 | [0.5 0.25 0.25] |
| bbtas | 2/3 | 18 | 6 | [0.0556 0.3333 0.3333 0.0556 0.0556 0.1667] |
| dk17 | 2/3 | 6 | 2 | [0.6667 0.3333] |
| donfile | 2/5 | 9 | 5 | [0.1667 0.1667 0.1667 0.1667 0.3333] |
| s400 | 3/21 | 12 | 3 | [0.6667 0.1667 0.1667] |
| s526 | 3/21 | 17 | 9 | [0.3333 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833] |

In Table 2, we report our results obtained when applying the KMS algorithm for a larger set of benchmarks. Once again, the input sequences were generated using Fibonacci series. For comparison, we also provide the results when the input is considered by default as having order one. For each case, the sequence-driven reachability analysis is carried out as above and then, using the $Q_{XS}$ matrix, the KMS algorithm is applied using a numerical partitioning criterion with a threshold of 0.001. In all cases, we report the number of (*input, state*) combinations reached and the number of reached states. The number of iterations needed to converge for an error less than $10^{-5}$ was $\leq 3$ for the second-order model and between 7 and 15 for the first-order model. The average run-time per iteration was 2 sec. for each block-matrix on a Sparc 20 workstation. For the larger sized matrices, the hierarchical KMS algorithm was used. For comparison, for first-order models we also provide the maximum and mean percentage errors obtained when comparing the results to the actual second order model (MAX% and MEAN%).

As we can see, considering the input of the target FSM as having only one-step temporal correlations may significantly impair the ability of predicting the correct number of reached states. In addition, for the subset of states correctly found as being reached, there is a significant error in the value of steady-state probabilities and total power consumption. For example, when excited using a second-order type of input, benchmark *planet* has a number of 34 reached states, whereas if the order is (incorrectly) assumed to be one, the number of reached states becomes 48. Moreover, the error in predicting the steady-state probability can be as high as 513% for the first-order model. Generally speaking, a lower order model covers all the states from the original one, but it may also introduce a significant number of extra states and, furthermore, the quality of the results in estimating the steady-state probabilities is seriously impaired.

Table 2: First-order vs. second-order model using KMS alg.

| | | Second-order | | | First-order | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Circuit | PIs/ FFs | #(*x*,*s*) | #*s* | Total power | #(*x*,*s*) | #*s* | MAX% | MEAN% | Total power |
| bbara | 4/4 | 28 | 3 | 747.38 | 86 | 10 | 49.40 | 35.91 | 786.53 |
| bbtas | 2/3 | 18 | 6 | 337.75 | 20 | 6 | 55.70 | 24.28 | 345.78 |
| dk17 | 2/3 | 6 | 2 | 1439.71 | 18 | 7 | 37.48 | 29.96 | 1313.78 |
| donfile | 2/5 | 9 | 5 | 3020.55 | 14 | 6 | 52.70 | 26.76 | 2943.15 |
| ex1 | 9/5 | 769 | 11 | 3165.66 | 1596 | 11 | 50.00 | 18.03 | 3082.86 |
| planet | 7/6 | 192 | 34 | 8550.92 | 3527 | 48 | 513.79 | 67.14 | 6429.16 |
| sand | 11/5 | 977 | 32 | 7883.13 | 17169 | 32 | 103.57 | 24.49 | 8162.71 |
| s1196 | 14/18 | 1536 | 329 | 7027.45 | 1918 | 342 | 133.71 | 4.17 | 7074.17 |
| s1238 | 14/18 | 1538 | 330 | 7755.56 | 1919 | 343 | 134.82 | 4.21 | 7648.88 |
| s1494 | 8/6 | 384 | 6 | 5367.42 | 1372 | 40 | 28.69 | 8.92 | 5082.92 |
| s526 | 3/21 | 17 | 9 | 1292.74 | 12155 | 4137 | 97.17 | 43.03 | 1342.12 |
| s820 | 18/5 | 680 | 8 | 4362.78 | 3056 | 24 | 24.78 | 10.78 | 4275.08 |

Since the values of these probabilities strongly affect the total power values, we also show the impact of these results on probabilistic power estimation techniques. Knowledge of the steady-state probability distribution for inputs and states and the conditional probabilities from matrix $Q_{XS}$, allow us to reduce the problem of power estimation for sequential circuits to the one for combinational circuits; therefore techniques like [19] can be successfully applied. We provide in the 'Total power' columns a comparison between the values of power estimated when the order of the input sequences was considered arbitrarily as being one vs. those determined when the actual order has been taken into account (all values are in µW at 20 MHz). As we can see, underestimating the actual order of the input sequence, strongly impacts the accuracy of the values of total power consumption; the error introduced can be as much as 25% for circuit *planet*.

## 7. Conclusion

In this paper we investigated from a probabilistic point of view the effect of finite-order statistics of the input sequence on FSMs behavior. In particular, the effect of temporal correlations longer that one clock-cycle was analyzed for steady-state and transition probabilities calculations. The results presented in this paper can be used in low-power design, synthesis and verification and represent an important step towards understanding the FSM behavior from a probabilistic point of view.


## Acknowledgments

The authors would like to thank Professor Carl Meyer from the Department of Mathematics of North Carolina State University for reading an early draft of the paper and providing valuable comments.


## References


[1] K. Parker and E.J. McCluskey, 'Probabilistic Treatment of General Combinational Networks', in *IEEE Trans. on Computers*, vol. C-24, June 1975.

[2] G. Hachtel, E. Macii, A. Pardo, and F. Somenzi, 'Probabilistic Analysis of Large Finite State Machines', in *Proc. ACM/IEEE Design Automation Conference*, pp. 270-275, June 1994.

[3] J. Savir, G.S. Ditlow, and P.H. Bardell, 'Random Pattern Testability', in *IEEE Trans. on Computers*, vol. C-33, Jan. 1984.

[4] G. Hachtel, M. Hermida, A. Pardo, M. Poncino and F. Somenzi, 'Reencoding sequential circuits to reduce power dissipation', in *Proc. ACM/IEEE Intl. Conf. Computer-Aided Design,* pp. 70-73, Nov. 1994.

[5] G.D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, 'Markovian Analysis of Large Finite State Machines', in *IEEE Trans. on CAD*, vol.15, no.12, Dec. 1996.

[6] C.-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain, and B. Lin, 'Power Estimation Methods for Sequential Logic Circuits', in *IEEE Trans. on VLSI Systems*, vol.3, no.3, Sept. 1995.

[7] J.G. Kemeny, J.L. Snell, and A.W. Knapp, 'Denumerable Markov Chains', D.Van Nostrand Company, Inc., 1966.

[8] A. Papoulis, 'Probability, Random Variables, and Stochastic Processes', McGraw-Hill Co., 1984.

[9] D. Marculescu, R. Marculescu, and M. Pedram, 'Stochastic Sequential Machine Synthesis Targeting Constrained Sequence Generation', in *Proc. ACM/IEEE Design Automation Conference*, pp. 696-701, June 1996.

[10] R. Marculescu, D. Marculescu, and M. Pedram, 'Adaptive Models for Input Data Compaction for Power Simulators', in *Proc. IEEE Asian-South Pacific Design Automation Conference*, Japan, Jan. 1997.

[11] R.E. Bryant, 'Graph Based Algorithms for Boolean Function Manipulation', in *IEEE Trans. on Computers*, vol. C-35, no. 8, pp. 677-691, Aug. 1986.

[12] O. Coudert, C. Berthet, and J.C. Madre, 'Verification of Sequential Machines Based on Symbolic Execution', in *Proc. of the Workshop on Automatic Verification Methods for Finite State Systems*, Grenoble, France, 1989.

[13] H.J. Touati, H. Savoj, B. Lin, R.K. Brayton, and A. Sangiovanni-Vincentelli, 'Implicit State Enumeration of Finite State Machines Using BDDs', in *Proc. ACM/IEEE Intl. Conf. Computer-Aided Design,* pp. 130-133, Nov. 1990.

[14] R. Koury, D.F. McAllister, and W.J. Stewart, 'Methods for Computing Stationary Distributions of Nearly-Completely-Decomposable Markov Chains', *SIAM Journal of Algebraic and Discrete Mathematics*, vol.5, no.2, pp. 164-186, 1984.

[15] W.J. Stewart, 'An Introduction to the Numerical Solution of Markov Chains', Princeton University Press, Princeton, 1994.

[16] C.D. Meyer, 'Stochastic Complementation, Uncoupling Markov Chains and the Theory of Nearly Reducible Systems', *SIAM Reviews*, vol.31, no.2, pp. 240-272, 1989.

[17] D. Marculescu, R. Marculescu, and M. Pedram, 'Sequence Compaction for Probabilistic Analysis of Finite State Machines', in *Proc. ACM/IEEE Design Automation Conference*, pp. 12-15, June 1997.

[18] H.A. Simon and A. Ando, 'Aggregation of variables in dynamic systems', *Econometrica*, vol.29, pp. 111-138, 1961.

[19] R. Marculescu, D. Marculescu, and M. Pedram, 'Efficient Power Estimation for Highly Correlated Input Streams', in *Proc. ACM/IEEE Design Automation Conference,* pp. 628-634, June 1995.