Composite Sequence Compaction for Finite-State Machines Using Block Entropy and High-Order Markov Models*

Radu Marculescu, Diana Marculescu, Massoud Pedram

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

Abstract - *The objective of this paper is to provide an effective technique for accurate modeling of the external input sequences that affect the behavior of Finite State Machines (FSMs). Based on the block entropy concept, we present a technique for identifying the order of variable-order Markov sources of information. Furthermore, using dynamic Markov modeling, we propose an effective approach to compact an initial sequence into a much shorter, equivalent one. The compacted sequence, can be subsequently used with any available simulator to derive the steady-state and transition probabilities, and the total power consumption in the target circuit. As the results demonstrate, large compaction ratios of orders of magnitude can be obtained without significant loss (less than 5% on average) in the accuracy of estimated values.*

I. 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. In particular, the behavior of FSMs has been investigated using concepts from the Markov chain theory. Studying then the behavior of the Markov chain provides us with different variables of interest of the original FSM. In this direction [1][2] are excellent references where steady-state and transition probabilities (as variables of interest) are successfully estimated in large FSMs. Both techniques are analytical in nature and, in order to manage complexity, have made some simplifying assumptions (temporal independence on the primary inputs being the most notable one). These assumptions, however, limit the applicability and usefulness of their results. As a consequence, only logic- or circuit-level simulation (under user-specified input vectors) produce the most accurate results. The downside is however the excessive simulation time needed. From this perspective, a short/compact sequence of stimuli - which is representative of the typical application data - would be desirable to speed-up the simulation.

Calculation of steady-state and transition probabilities is a very important topic by itself, because these probabilities completely characterize the behavior of a FSM. However, a particular domain where this calculation has an immediate application is the power estimation area. Without loss of generality, we will consequently emphasize the applicability of the new results to sequence compaction for simulation-based power evaluation. The problem of producing a shorter sequence (compared to a given one) which can preserve the set of steady-state and transition probabilities can be now cast in terms of power as follows: can we transform a given input sequence into a shorter one, such that the new body of data captures statistics which completely determine the power dissipation in a circuit?

Generating a minimal-length sequence of input vectors that satisfies a prescribed set of statistics in not a trivial task. The reason is the elaborate set of input statistics that must be preserved or reproduced during sequence generation for use by power simulators. Elaborate and effective techniques were recently presented in [3] [4] where the authors succeed in compacting large sequences with very small loss in accuracy. However, these approaches are suited only for combinational circuits; this is because both of them consider only first-order temporal effects, that is they analyze only pairs of consecutive vectors to perform sequence compaction. As it can be proven, in the case of FSMs, this is not enough for accurate estimation of transition probabilities. Temporal correlations longer than one time step can affect the overall behavior of the FSM and therefore, result in very different power consumptions. Let us illustrate this point using a simple example.

__Example 1__: Let S_1 and S_2 be two 2-bit sequences, of length 48, as shown in Fig.1a. These two sequences, have exactly the same set of first-order temporal statistics, that is, they cannot be distinguished as far as wordwise one-step transition probabilities are concerned. In fact, in Fig.1b we provide the wordwise transition graph for these two sequences. Each node in this graph is associated to a distinct pattern that occurs in S_1 and S_2 (the upmost bit is the most significant one, e.g. in S_1, v_1 = '1', v_2 = '2'). Each edge represents a valid transition between any two valid patterns and has a nonzero probability associated with it.

S_1

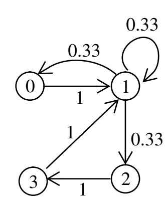
v_{48} ... $v_3 v_2 v_1$

011011011000110000001101100000000000110000000110
110110110111101101011011010111010111010110101101

S_2

001100001100001100001100001100001100001100001100
011011011011011011011011011011011011011011011011

(a)



(b)

Fig.1

Starting with different initial states and using a random number generator we may, of course, generate other sequences equivalent with S_1 and S_2 as far as the one-step transition probabilities are concerned. We can then see the graph in Fig.1b as a compact, canonical, characterization of sequences S_1 and S_2. Suppose now that S_1 and S_2 are fed into the benchmark circuit *dk17*. Looking at different internal nodes of the circuit, we observe that the total number of transitions made by each node is very different when the circuit is simulated with S_1 or S_2. Moreover, the total power consumption at 20 MHz is 1298.18 μW and 1429.85 μW, respectively, showing a difference of about 10% even for this small set of inputs.

A natural question is then, why does this difference

*This research was supported in part by DARPA under contract F33615-95-C1627, SRC under contract 97-DJ-559, and NSF under contract MIP-9457392.

appear, despite the fact that S_1 and S_2 have the same characteristic graph shown in Fig.1b? The reason resides in the fact that S_1 and S_2 have a different set of second-order statistics that is, the sets of triplets (three consecutive patterns) are different for these sequences. For instance, the triplet (0, 1, 2) in S_1 does not occur in S_2; the same observation applies to the triplet (0, 1, 0) in S_1. For now, two observations can be made:

- knowing the *order* of the temporal correlations is important because, having the same set of one-step transition probabilities *does not* mean that the sets of second-(or higher) order statistics are identical;
- for FSMs, second- (or higher) order statistics can make a significant difference in total power consumption.

Addressing these issues, the present paper improves the state-of-the-art in three ways: first, it shows the effect of high-order temporal statistics of the input sequence on the FSM behavior; second, based on the block entropy concept, it introduces a technique for identifying the order for *composite* sources of information (that is, sequences that can be piecewise modeled by Markov chains of different orders); finally, based on the vector compaction paradigm, it provides an original solution to the power estimation problem for FSMs.

To conclude, this research is beneficial to not only simulation-based and probabilistic approaches for power estimation, but also the general FSM analysis techniques relying on probabilistic premises.

The paper is organized as follows: in Section II we present the main results about the effect of finite-order statistics on FSM behavior and, based on the block entropy concept, we introduce a technique for identifying the actual order of a Markov source. Section III discusses the issue of data compaction for solving the power estimation problem. Section IV gives a practical procedure for sequence compaction and experimental results for common sequential benchmarks. Finally, we conclude by summarizing our main contribution.

II. High-Order Markov Sources Of Information

In what follows, we model the input sequences as binary information sources of discrete Markov type that emit symbols at each time step, according to some probabilistic rule. We also restrict our investigation to the case of stationary and ergodic sources of information and analyze the effects of input statistics on FSM behavior.

A. Finite-order Markov chains

First, we focus on the input sequence that feeds a target circuit and we model it as an information source.Without loss of generality, we restrict ourselves to the set S of all finite binary sequences on b bits. A particular sequence S_1 in S consists of vectors $v_1, v_2,..., v_n$ (which may be distinct or not), each having a positive occurrence probability. Indices 1, 2,..., n represent the discrete time steps when a particular vector that occurs in the sequence is applied to the target circuit.

An important class of information sources is the class of Markov sources which can be conveniently modeled as Markov chains of finite-order.

Definition 1. (lag-k Markov chain). A discrete stochastic process $\{v_n\}_{n\geq 1}$ is said to be a lag-k Markov chain if at any time step $n \geq k+1$:

$$p(v_n|v_{n-1}v_{n-2}\ldots v_1) = p(v_n|v_{n-1}v_{n-2}\ldots v_{n-k}) \qquad (1)$$

In particular, any lag-one Markov source, is characterized by the set of internal states (nodes in the corresponding graph representation) and the set of one-step conditional probabilities q_{ij} that give the transition probability from state v_i to the next state v_j. It should be noted that any lag-k Markov chain can be reduced to a lag-one Markov chain based on the following result.

Proposition 1. [5] If $\{u_n\}_{n\geq 1}$ is a lag-k Markov chain then $\{v_n\}_{n\geq 1}$ where $v_n = (u_n, u_{n+1},..., u_{n+k-1})$ is a multivariate first-order Markov chain.

B. The effect of finite-order statistics of the input sequence on FSM behavior

Now we turn our attention from the input sequence to the circuit and investigate the effect of input statistics on the transition probabilities (for primary inputs and present state lines) in the target circuit. As shown in Fig.2, x_n, s_n will denote the inputs and states of the target sequential machine; $p(x_n s_n)$ is the probability that the input is x_n and the state is s_n at time step n. As usual, *'next'* denotes the next state function of the FSM and *'out'* represents the output function.

We are interested in defining the joint probabilities $p(x_n s_n)$ and $p(x_n s_n x_{n-1} s_{n-1})$ because, as we can see in Fig.2, they completely capture the characteristics of the primary inputs and present state lines that feeds the next state and the output logic of the target circuit.

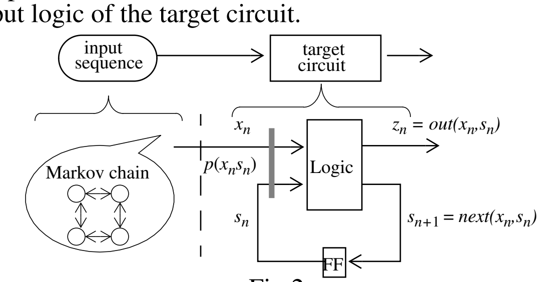


Fig.2

Under the general assumption of stationarity and ergodicity we can prove the following result.

Theorem 1.[1] If the sequence feeding a target sequential circuit has order k, then a lag-k Markov chain which correctly models the input sequence, also correctly models the k-step conditional probabilities of the primary inputs and internal states in the target circuit, that is $p(x_n s_n|x_{n-1}s_{n-1}x_{n-2}s_{n-2}\ldots x_{n-k}s_{n-k}) = p(x_n|x_{n-1}x_{n-2}\ldots x_{n-k})$.

We note therefore that preserving the order-k statistics for the primary inputs implies that order-k statistics will also be captured for inputs and states. However, modeling an order-k source by a lower order source may introduce large inaccuracies as shown in the next example.

<u>Example 2</u>*:* We consider once again the sequences S_1 and S_2 that feed the benchmark *dk17* and illustrate that indeed, if the input sequence has order two, then modeling it as a lag-one Markov Chain will *not* preserve the first-order joint transition probabilities (primary inputs and internal states) in the target circuit. We simulated the benchmark *dk17 (*starting with the same initial state *'19')* for both sequences and we present in Fig.3 (Fig.3a is for S_1 and Fig.3b is for S_2) the wordwise transition graphs obtained for the signal lines $(x_n s_n)$.

[1]All proofs can be found in [9].

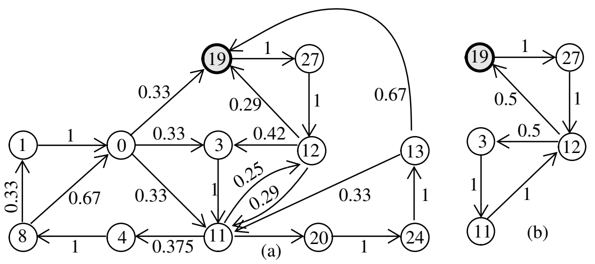


Fig.3

The benchmark *dk17* has 2 primary inputs and 3 FFs therefore in Fig.3, any node is decimally encoded using 5 bits. For instance, the initial state *'19'* corresponds to the binary code '10011' that is, '10' for primary inputs and '011' for state lines. Applying '11' on the primary inputs, the present state becomes '011', therefore we enter the node '11011' = 27 in Fig.3. As we can see, because S_1 can be modeled as a first order Markov source, while S_2 must be modeled as a second order Markov source, the corresponding transition graphs in Fig.3 are quite different. From a practical point of view, this means that if one underestimates a high-order source (for instance, assuming that second- or higher-order temporal correlations are not important), then one may end up not preserving even the first-order transition probabilities in the target circuit. In terms of power consumption, this adversely affects the quality of the results as shown in Example 1.

C. The order of a Markov source

Now, we try to determine the order of a Markov source, because, as proved in Theorem 2, the knowledge of the correct value of k, is essential in FSM analysis. To this end, based on the probability of finding a 'block' of vectors $\langle v_n, v_{n-1}, \ldots, v_1 \rangle$[1] in any sequence in S (denoted by $p(v_n v_{n-1} \ldots v_1)$), we introduce the following entropy-like quantities.

Definition 2. (*block entropy)* The *block entropy* of length n is defined as:

$$H_n = \sum_{\langle v_n v_{n-1} \ldots v_1 \rangle} p(v_n v_{n-1} \ldots v_1) \cdot \log p(v_n v_{n-1} \ldots v_1) \qquad (2)$$

where $n \geq 1$.

Definition 3. (*conditional entropy)* The *conditional entropy* associated with the addition of a new vector v_{n+1} to the left of an existing block $\langle v_n, v_{n-1}, \ldots, v_1 \rangle$ is defined as:

$$h_n = H_{n+1} - H_n \qquad (3)$$

for $n \geq 1$ and $h_0 = H_1$.

Definition 4. (*source entropy)* The *entropy of a source,* or the *uncertainty per step,* is defined as:

$$h = \lim_{n \to \infty} \frac{H_n}{n} = \lim_{n \to \infty} h_n \qquad (4)$$

and often referred to as *metric entropy* or *entropy rate* [5].

For stationary and ergodic processes, Khinchin and McMillan [6] have shown that H_n in (2) is monotonically increasing, H_n/n is monotonically decreasing and the limit in (4) exists. It can also be shown that the conditional entropy h_n is a measure of the predictability for the whole process.

[1]Because the process is stationary and ergodic, the lower indices 1, 2,... n do not designate the absolute time instances, but the sequencing among the vectors that occur within a block of length n.

Lemma 1. For any lag-k Markov chain, it holds that $H_n = H_k + (n-k) \cdot (H_{k+1} - H_k),\ \forall n \geq k$.

Due to statistical correlations extending over a range of only k iterations, for Markov sources of order k, the conditional uncertainty h_n is decreasing [5] until it eventually reaches its limit value h for $n = k$. The memory effects of the source are reflected by the *saturation point* that is, that value of n when the limit h is reached exactly or with a good approximation.

<u>*Example 3:*</u> In Fig.4 we have the typical behavior for a lag-one and a lag-two Markov sequences, which have been generated by using a first and a second order recursive relations, respectively. The two sources have the same order 0 and 1 statistics, but different order-2 statistics. In both cases, the n-step conditional entropy h_n of the source reaches its limit h after few tens of vectors; in the first case, the limit is $h = h_1$, whereas in the case of the lag-two Markov source, the limit is given by h_2 as it was expected according to the above result.

In practice, lag-k Markov chains with fixed k may not be a good approximation for a given source. For example, the sequence can be a mixture of subsequences, each generated from a different order Markov source. We call such a sequence a *composite sequence* to emphasize its non-homogeneous characteristics. When the sequence changes its behavior due to the change in order, the stationarity and convergence hypotheses from (4) do not hold anymore. For instance, if we consider two mixed sequences, the first containing a lag-one followed by a lag-two Markov sequence, and the second a lag-two followed by a lag-one Markov sequence, the behavior of conditional entropies is the one depicted in Fig.5.

In the first case, after a sufficiently large number of steps, h_1 and h_2 are no longer the same. Actually, in the long run, the limit in (4) becomes h_2, thus detecting an order 2 for the source. In this case, the convergence to h_1 is violated and thus the sequence changes its order from 1 to 2. In the second case, after having already stabilized to the stationary values, the conditional entropies h_2, h_3, h_4,..., tend to increase so that the order of the second half of the sequence can no longer be considered 2.

This discussion provides a starting point for determining the order of subsequences in a dynamic fashion. The only requirement that has to be satisfied is the stationarity of each subsequence and in this case, the entire sequence is called *piecewise stationary.* This is the basic hypothesis for all our experiments.

III. Data Compaction For Power Estimation

So far, we have characterized the family of high-order Markov sources and, under stationarity and ergodicity hypothesis, we have presented an alternative to identify their order using the block entropy concept. In the present section, we investigate the issue of data compaction for power estimation in sequential machines and analyze conditions in which the statistics of the original sequence can be preserved throughout the compaction process.

A. Problem formulation

Assuming that a gate level implementation is available, to estimate the total power dissipation, one can sum over all the gates in the circuit the average power dissipation due to the capacitive switching currents, that is:

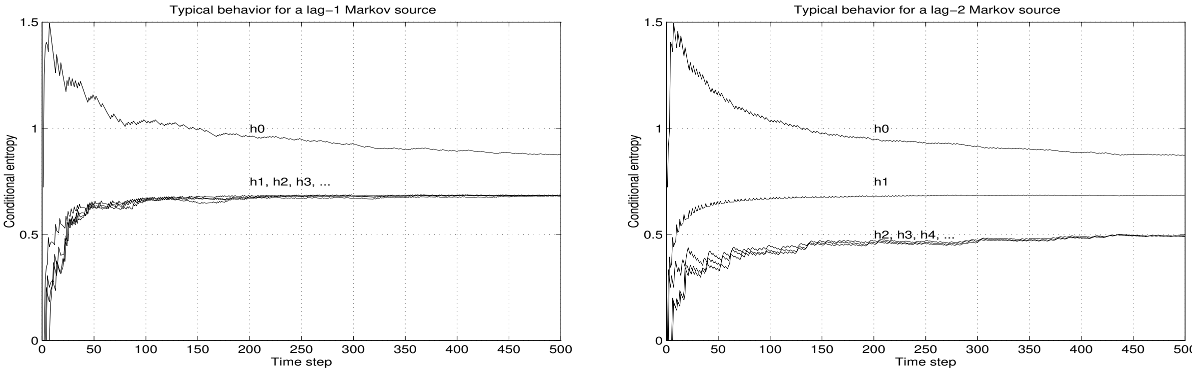


Fig.4

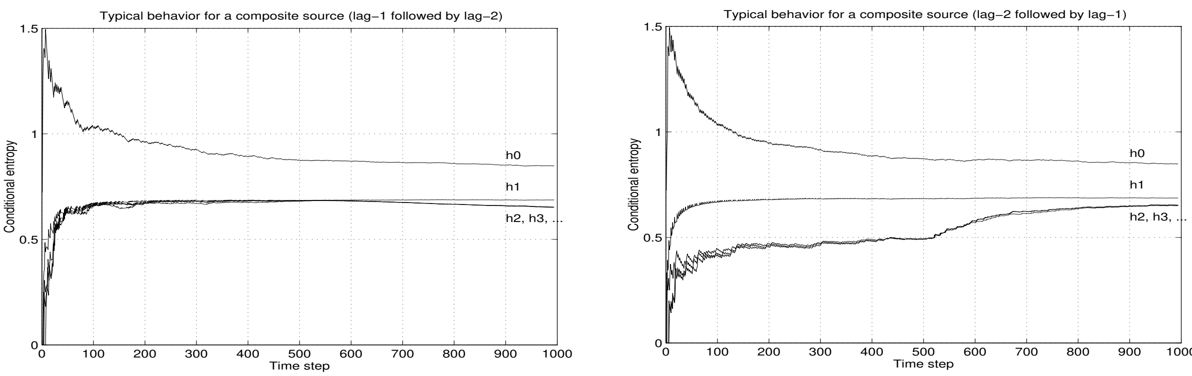


Fig.5

$P_{avg} = 0.5 \cdot f_{clk} \cdot V_{DD}^2 \cdot \sum_n (C_n \cdot sw_n)$ where f_{clk} is the clock frequency, V_{DD} is the supply voltage, C_n and sw_n are the capacitance and the average switching activity of gate n, respectively. From here, it becomes clear that the average switching activity per node (gate) is the key parameter that needs to be correctly determined, if we are interested in a node-by-node power estimation. However, this parameter is highly sensitive to the input statistics, namely it greatly depends on transition probabilities among different signal lines. As shown in the previous section, high-order information sources make a significant difference in power consumption for sequential machines.

Having these issues in mind, the *vector compaction problem* for FSMs can be formulated as follows: for any sequence of length L_0, find another sequence of length $L \ll L_0$ (consisting of a subset of vectors from the original sequence), such that the average joint transition probability on the primary inputs and present state lines is preserved *wordwise,* for k consecutive time steps. More formally, the following holds:

$$\left| \begin{matrix} p(x_n s_n x_{n-1} s_{n-1} \ldots x_{n-k} s_{n-k}) - \\ - p'(x_n s_n x_{n-1} s_{n-1} \ldots x_{n-k} s_{n-k}) \end{matrix} \right| < \varepsilon \qquad (5)$$

where p and p' are the probabilities in the original and compacted sequences, respectively. This condition simply requires that the joint transition probability for inputs and states $(x_i s_i)$ is preserved within a given level of error for k consecutive time steps.

B. Statistics of the original and compacted sequences

Without loss of generality, we will refer subsequently only to first-order Markov chains. However, by virtue of Proposition 1, all results easily translate to lag-k Markov chains. We also assume cardinalities l and m for the input and reachable state spaces, respectively, and introduce a new formulation based on matrices because this is a convenient notation for our purpose.

The first-order Markov chain at the primary inputs x_n of the FSM in Fig.2, can be characterized by the matrix of conditional probabilities $Q_X = (q_{ij})_{1 \le i,j \le l}$, where $q_{ij} = p(x_n = v_j | x_{n-1} = v_i)$ and $\{v_1, v_2, \ldots, v_l\}$ represents the whole set of possible vectors at the primary inputs. On the other hand, the Markov chain defined jointly for the primary inputs and state lines (x_n, s_n) can be characterized by the matrix $Q_{XS} = (r_{ip,jq})_{\substack{1 \le i,j \le l \\ 1 \le p,q \le m}}$, where $r_{ip,jq} = p(x_n s_n = v_j u_q | x_{n-1} s_{n-1} = v_i u_p)$ and $\{u_1, u_2, \ldots, u_m\}$ represents the set of reachable states.

Proposition 2. The matrix Q_{XS} can be written in the following form:

$$Q_{XS} = \begin{bmatrix} q_{11}B_1 & q_{12}B_1 & \ldots & q_{1l}B_1 \\ q_{21}B_2 & q_{22}B_2 & \ldots & q_{2l}B_2 \\ \ldots & \ldots & & \ldots \\ q_{l1}B_l & q_{l2}B_l & \ldots & q_{ll}B_l \end{bmatrix} \qquad (6)$$

where $\{B_i\}_{1 \le i \le l}$ is a set of $m \times m$ degenerate[1] stochastic

[1] That is, the elements of the matrices B_i are only 0 or 1.

matrices defining the next state function for input v_i; specifically, if $B_i = (b^i_{pq})_{1 \le p,q \le m}$ then

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

Lemma 2. If matrix Q_X is preserved (during the compaction process) up to an ε > 0, then matrix Q_{XS} is also preserved up to the same ε.

Theorem 2. If matrix Q_X is preserved up to an ε > 0, then steady-state probabilities for the primary inputs and state lines are also preserved up to a polynomial function of ε having the form $0(\varepsilon) = k_1\varepsilon + k_2\varepsilon^2 + \dots$.

Corollary 1. If matrix Q_X is preserved up to an ε > 0, then transition probabilities for the primary inputs and state lines for two consecutive time steps are asymptotically close to the original ones, that is $\left|p(x_n s_n x_{n-1} s_{n-1}) - p'(x_n s_n x_{n-1} s_{n-1})\right| = 0(\varepsilon)$.

IV. Practical Considerations And Experimental Results

As a conclusion from sections II and III, what we need is an efficient way to model lag-*k* Markov chains that might characterize the input sequences that feed the target circuit. For this purpose, we choose the Dynamic Markov Chain (DMC) technique used recently to adaptively compact data for power simulators [7].

The structure DMT_1 used by authors in [7] is general enough to capture completely the correlations among all bits of the same input vector and also between successive input patterns. However, it has conceptually no inherent limitation to be further extended to capture temporal dependencies of higher orders. For instance, if we continue to define recursively DMT_2 (as a function of DMT_1), we can basically capture second-order temporal correlations. For any sequence where v_i, v_j, v_k are three consecutive vectors (that is, $v_i \rightarrow v_j \rightarrow v_k$), the tree DMT_2 looks like in Fig.6.

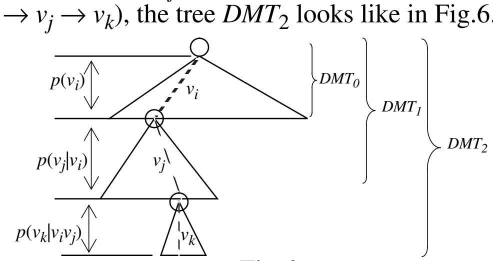


Fig.6

The following result, gives the theoretical basis for using the DMC technique to capture high-order temporal correlations.

Theorem 3. The general structure DMT_k and its parameters completely capture spatial and temporal correlations of order *k*.

The DMC modeling approach offers the significant advantage of being a *one-pass adaptive technique*. As a one-pass technique, there is no requirement to save the whole sequence in the on-line computer memory. Starting with an initial empty tree DMT_k, while the input sequence is scanned incrementally, both the set of states and the transition probabilities change dynamically making this technique highly adaptive. Also, using this data structure, we can easily account for conditional entropies and detect the order of the Markov source. Under stationarity conditions, the order is detected as the minimum *k* such that $|h_k - h_n| < \varepsilon$, for some ε > 0 and any $n = k + 1, \dots, K$ where *K* is the maximum order of the source to be detected. After that, if either this condition becomes violated or the stationarity hypothesis doesn't hold, the model is flushed and restarted. For each grown tree, the generation phase is driven by the user-specified compaction parameter *ratio,* that is, we generate a total of *m* = *n*/*ratio* vectors. We also note that this strategy does note allow 'forbidden' vectors, i.e., those combinations that did not occur in the original sequence, will not appear in the final compacted sequence either. This is an essential capability needed to avoid 'hang-up' states of the sequential circuit during simulation process for power estimation.

Starting with a *k*-bit input sequence of length *n*, we perform a one-pass traversal of the original sequence and simultaneously build the basic tree DMT_k; during this process, the frequency counts on DMT_k's edges are dynamically updated.

The overall strategy is depicted in Fig.7.

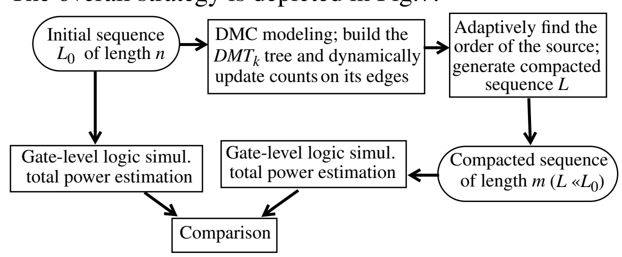


Fig.7

The next step in Fig.7 does the actual generation of the output sequence (of length *m*). To generate the new sequence we use a modified version of the dynamic weighted selection algorithm presented in [8]. If the initial sequence has length *n* and the new generated sequence has length *m* < *n,* then the outcome of this process is a compacted sequence; we say that a *compaction ratio* of *r* = *n*/*m* was achieved.

Finally, a validation step is included in the strategy; we have used an in-house gate-level logic simulator developed under SIS. The total power consumption of some sequential benchmarks has been measured for the initial and the compacted sequences, making it possible to assess the effectiveness of the compaction procedure.

In Table 1, we provide the gate-level power simulation results for different initial sequences having the total length of 10,000 vectors; these sequences were produced using a second order information source based on Fibbonacci series. As shown in Table 1, the sequences were compacted with two different compaction ratios (*r* = 10 and 20) using three Markov models: one of order zero (that is, assuming temporal independence on the primary inputs), one of order one (that is, based on DMT_1) and another one matching the characteristics of the sequence (that is, based on DMT_2). We give in this table the total power dissipation measured for the initial sequence (column 3) and for the compacted sequence using all models (columns 4-9). On a Sparc 20 workstation with 64 Mbytes of memory, the time necessary to read and compress data was less than 10 seconds for all models. Since the compaction with DMC modeling is linear in the number of nodes in the DMT_k structure, these time values are far less than the actual time needed to simulate the whole sequence. During these experiments, the maximum number of nodes allowed in the Markov model was 200,000.

Table 1: Total Power (μW@20MHz) for input sequences of order 2

			Power for r = 10			Power for r = 20		
Circuit	PIs/FFs	Power for init. seq.	Order 0	Order 1	Adaptive	Order 0	Order 1	Adaptive
s1196	14/18	7025.31	6740.16	6668.31	7027.58	6667.81	6414.11	7006.11
s1423	17/74	5624.64	5197.43	5203.94	5577.11	5068.73	5084.82	5458.20
s510	19/6	2073.55	2210.85	1940.11	1980.47	2277.32	1877.72	1974.88
s5378	35/164	13826.55	11955.22	13304.01	13666.80	11885.03	12838.12	13684.05
s820	18/5	4120.95	3864.65	3668.70	4077.06	3703.84	3561.86	4047.21
s9234	36/211	12531.45	13004.71	13037.10	12247.65	13239.04	13070.32	12187.86
s953	16/29	1158.04	1199.03	1122.11	1156.47	1207.44	1102.16	1149.19
		Avg. % err.	6.47	5.84	1.43	8.41	8.24	2.04
bbara	4/4	747.25	689.54	821.74	745.57	685.09	823.67	745.67
bbtas	2/3	337.74	284.40	362.00	337.43	284.15	323.36	337.39
dk17	2/3	1439.65	1203.75	1306.32	1437.21	1203.56	1286.46	1436.34
donfile	2/5	3020.63	2242.27	2874.66	3016.54	2207.00	2852.06	3010.74
mc	3/2	295.99	234.90	238.07	293.32	219.47	246.30	292.79
planet	7/6	8552.95	7314.99	6505.96	8442.30	7105.18	6565.44	8151.24
shiftreg	1/3	144.60	91.43	124.27	144.49	91.27	122.56	144.63
		Avg. % err.	19.65	12.69	0.41	21.03	12.29	0.95

As we can see, for the model of order 2, the quality of results is very good even when the length of the initial sequence is reduced by more than one order of magnitude. Thus, for *s1196* in Table 1, instead of simulating 10,000 vectors with an exact power of 7025.31 μW, one can use only 1,000 vectors (r = 10) with an estimate of 7027.88 μW or just 500 vectors (r = 20) with power consumption estimated as 7006.11 μW. This reduction in the sequence length has a significant impact on speeding-up the simulative approaches where the run time is proportional to the length of the sequence which must be simulated. On the other hand, using a zeroth- or first-order model, the quality of the results can be seriously impaired. For instance, in the case of benchmark *plane*t, for r = 20, we can erroneously predict a total power of 6505.96 and, for r = 10, a value of 6565.24 for a first-order model (more than 23% error). The value of the error can be even worse for a zero-order model (e.g. benchmark *donfile,* where the error is more than 25%). This is because for a sequence generated with a second-order source, a model that ignores temporal correlations or considers only pairs of consecutive vectors cannot preserve correctly even the first-order transition probabilities for the primary inputs and state lines.

Finally, we give in Table 2 the results obtained for composite sequences of length 10,000. These sequences have been generated using different generators and exhibit temporal correlation of various orders (order 2, followed by order 1 and finally, once again order 2). This hybrid character of the sequences makes a significant difference in terms of total power consumption for the analyzed benchmarks. As we can see, compared to Table 1, the most dramatic increase in the level of error occurs for the zeroth-order model which shows that the temporal independence assumption is impairing the accuracy of the estimates. The error for the first-order model is on average around 10%, while the adaptive modeling technique provide accurate results, even for a compaction ratio of r = 20. Looking at the results, we can therefore conclude that only the adaptive technique is appropriate for correctly modeling the input sequence.

Table 2: Total Power (μW@20MHz) for composite input sequences

			Power for r = 10			Power for r = 20		
Circuit	PIs/FFs	Power for init. seq.	Order 0	Order 1	Adaptive	Order 0	Order 1	Adaptive
s1196	14/18	5272.36	7212.58	5395.73	5237.65	7245.41	5501.05	4972.52
s1423	17/74	3964.48	5771.69	4173.21	4048.88	5836.40	4046.63	3834.35
s510	19/6	1520.46	2232.08	1814.65	1527.91	2215.03	1953.39	1501.10
s5378	35/164	11566.16	14251.17	11711.97	11282.86	14325.13	11871.03	10990.86
s820	18/5	3131.51	4158.82	3368.83	3096.87	4178.75	3431.04	3196.87
s9234	36/211	10046.05	12797.24	9688.30	9573.48	12951.02	9672.07	9288.65
s953	16/29	764.92	1188.94	796.02	755.66	1186.26	805.89	731.35
		Avg. % err.	38.29	6.20	1.82	38.80	8.04	4.18
bbara	4/4	771.16	790.23	842.87	769.97	806.44	749.96	766.41
bbtas	2/3	405.29	333.21	427.52	403.45	339.82	443.14	402.84
dk17	2/3	1392.67	1165.08	1258.88	1390.21	1135.61	1248.20	1388.09
donfile	2/5	3203.97	2455.90	3100.94	3197.36	2463.37	2997.86	3190.14
mc	3/2	335.97	289.37	298.76	332.92	295.64	285.17	328.27
planet	7/6	8619.93	7753.93	7234.10	8401.28	7605.67	7097.61	8030.61
shiftreg	1/3	149.59	100.62	140.30	149.29	97.36	139.68	148.97
		Avg. % err.	16.65	8.71	0.66	17.28	9.76	1.65

V. 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. Based on the block entropy concept, we presented a technique for identifying the order of variable-order Markov sources of information. Finally, using dynamic Markov modeling, we proposed an effective approach to compress an initial sequence into a much shorter one such that the steady state and transition probabilities (and therefore the total power consumption) in the target circuit are preserved.

For the first time to our knowledge, the effect of temporal correlations longer that one clock-cycle on the power dissipation in FSMs was studied. The results presented in this paper represent an important step towards understanding the FSM behavior from a probabilistic point of view.

REFERENCES

[1] 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.

[2] 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.

[3] C.-Y. Tsui, R. Marculescu, D. Marculescu, and M. Pedram, 'Improving the Efficiency of Power Simulators by Input Vector Compaction', in *Proc. ACM/IEEE Design Automation Conference*, pp. 165-168, June 1996.

[4] 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.

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

[6] Khinchin, A. I., 'Mathematical Foundations of Information Theory', Dover Publisher, New York, 1957.

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

[8] J.W.Green and K.J.Supowit, 'Simulated Annealing without Rejected Moves', in *Digest. of Intl. Conference on Computer Design,* pp. 658-663, Oct. 1984.

[9] R. Marculescu, D. Marculescu, and M. Pedram, 'Compositie Sequence Compaction for FSMs', Technical Report, CENG 97, Univ. of Southern California.