Compacting Sequences with Invariant Transition Frequencies

ALI PINAR
Lawrence Berkeley National Laboratory

and

C. L. LIU
National Tsing Hua University

Simulation-based power estimation is commonly used for its high accuracy despite excessive computation times. Techniques have been proposed to speed it up by compacting an input sequence while preserving its power-consumption characteristics. We propose a novel method to compact a sequence that preserves transition frequencies. We prove the problem is NP-complete, and propose a graph model to reduce it to that of finding a heaviest-weighted trail, and a heuristic utilizing this model. We also propose using multiple sequences for better accuracy with even shorter sequences. Experiments show that power dissipation can be estimated with an error of only 2.3%, while simulation times are reduced by 10. Proposed methods generate solutions that effectively preserve transition frequencies and that are very close to optimal. Experiments also show that multiple sequences grant more accurate results with even shorter sequences.

Categories and Subject Descriptors: B.7.2 [**Hardware**]: Integrated Circuits—*design aids*; F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*nonnumerical algorithms and problems*; G.2.2 [**Mathematics of Computing**]: Discrete Mathematics—*graph theory*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Sequence compaction, power estimation, graph algorithms

1. INTRODUCTION

The growing need for low-power systems requires accurate power estimation. Simulating a circuit for a typical vector sequence gives very accurate results, but requires excessive computation times. Besides, results are highly dependent on the input vector sequence. To alleviate such dependency, lengths of the input sequences should be very long, leading to excessive simulation times. For a comprehensive cov-

This work is supported in part by the National Science Foundation (NSF) under grant 1-5-31333 NSF MIP 96 12184. A. Pınar is also supported by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract DE-AC03-76SF00098.
Authors' addresses: A. Pınar Lawrence Berkeley Lab, One Cyclotron Road MS 50F, Berkeley, CA 94720. email:apinar@lbl.gov; C. L. Liu, Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. email: liucl@mx.nthu.edu.tw.

erage of the power estimation literature, the survey by Pedram [1999] is a valuable resource. For a CMOS circuit, dominant source of power dissipation is the dynamic transition current. In combinational circuits, power consumption corresponding to an input vector sequence depends only on transitions between successive input vectors. Thus if a given vector sequence can be transformed to a shorter one with invariant transition frequencies, the shorter sequence can be used for power estimation. This problem is called the *Sequence Compaction* problem, and has been studied recently [Tsui et al. 1996; Huang et al. 1996; Marculescu et al. 1997b; 1997a; Pınar and Liu 1998].

This article investigates combinatorial aspects of the sequence compaction problem. We prove the problem is NP-complete, and propose a graph model to reduce it to that of finding a heaviest-weighted trail, and describe a heuristic to find such trails. In our model, each distinct vector of the sequence defines a vertex, and each transition defines parallel edges, where the multiplicity of edges and weight of each edge depends on the frequency of this transition. We also discuss generating multiple sequences with different compaction factors, as opposed to merely a single sequence, for better accuracy with even shorter sequences. Experiments verify that simulation times can be significantly reduced with highly accurate results. Error in estimations is limited to only 2.3%, while the simulations are 10 times faster. We also show that transition frequencies are very accurately preserved and multiple sequences grant better accuracy with even shorter simulation times.

A more detailed description of our work can be found in [Pınar and Liu 2002].

2. SEQUENCE COMPACTION PROBLEM

Sequence compaction aims at transforming a sequence to a shorter one with similar characteristics. Here we work on sequences of binary input vectors of a circuit and want to preserve power characteristics. A compacted sequence preserves power characteristics of the original, if it "moves" all internal states of the circuit with the same frequency as the original. The major source of power dissipation in a circuit is the switching activity, which is triggered by transitions in the input vectors. If a sequence can be compacted into a shorter one with the same vector transition frequencies, the compacted sequence can be used for power estimation. Preserving bit-transition frequencies, as opposed to vector-transition frequencies, or decomposing vectors into highly correlating bit groups and preserving frequencies of these groups have been proposed [Marculescu et al. 1997b]. In this work, we address preserving frequencies of vector transitions.

An input sequence $S = \langle s_1, s_2, \ldots, s_m \rangle$ is a sequence of binary n-vectors. A *transition* $t = (s_i, s_j)$ is an ordered pair of distinct n-vectors. We use $S(t)$ to denote the number of transitions t, and $T(S)$ to denote the set of transitions in sequence S. Marculescu *et al.* [1999] prove that satisfying the condition $|p(t) - p'(t)| < \epsilon$ for all transitions, limits the error in estimation to $O(\epsilon)$, where ϵ is an infinitesimal quantity and $p(t)$ and $p'(t)$ denote the transition probabilities of transition t in the original and compacted sequences, respectively. We define the sequence compaction problem as follows.

Given a compaction factor c, and an input sequence $S = \langle s_1, s_2, \ldots, s_m \rangle$, construct *a new sequence S' to minimize the cost $\mathcal{C}(S, S', c)$, where*

$$\mathcal{C}(S, S', c) = \sum_{t \in T(S)} \frac{|S(t) - cS'(t)|}{S(t)}. \tag{1}$$

This formula preserves frequencies of all transitions, as advised in [Marculescu et al. 1999]. It is worth noting that our techniques can be used with alternative cost functions as described in Section 2.2.

2.1 Complexity of Sequence Compaction

We prove that the sequence compaction problem is NP-complete by reduction from the Hamiltonian path problem [Cormen et al. 1990].

THEOREM 2.1. *Given a compaction factor c, an input sequence $S = \langle s_1, \ldots, s_m \rangle$, and a bound B, deciding if a sequence S' exists so that $\mathcal{C}(S, S', c) \leq B$ is NP-complete.*

PROOF. Let $G = (V, E)$ be a directed graph. Let $M, N \geq 4$ be arbitrarily large and even integers. For each vertex $v_i \in V$, repeat subsequence $v_{i1}, \ldots, v_{iN}$ M times and add x_i at the end. And for each edge $(v_i, v_j) \in E$, repeat the subsequence v_{iN}, v_{j1}, e_{ijk} $M/2$ times, each time with a different k value. S is a concatenation of all subsequences followed by a sentinel y. In this construction, dummy variables: x_i, e_{ijk}, separate vertex- and edge-sequences to encode the sets in a sequence. The proof is based on the following remarks. Compact this sequence by a factor of M. First, if a transition is not in the original, then it cannot be in the compacted sequence, since its cost is ∞ according to Equation 1, thus a subsequence for a vertex either appears as a whole, or does not appear at all. Second, if a vertex subsequence $v_{i1}, \ldots, v_{iN}$ occurs just once, the penalty will be zero. Each deviation from one adds a cost of $N-1$. Absence of $v_{iN}v_{i1}$ transitions costs 1. Minimum cost due to vertex subsequences is $|V|$, achieved when each vertex sequence occurs exactly once. Third, for each edge transition v_{iN}, v_{j1}, zero or one appearance costs the same: 1. After the first, each extra appearance costs 2. Thus minimum total cost for edges is $|E|$, achieved when each edge transition occurs at most once. And finally, dummy variables e_{ijk} and x_i appear in two transitions, and absence of these variables costs only 2, whereas their presence costs $2(M-1)$ for each, thus the minimum total costs for e_{ijk} and x_i are $M|E|$ and $2|V|$, respectively.

The second remark implies that the cost of S' is minimum when it includes all vertex subsequences exactly once. Moreover, these subsequences must be connected by edge transitions, because by the first remark, the compacted sequence does not have any transitions not in the original sequence, and a transition (v_{iN}, v_{j1}) occur in the sequence only if there is an edge from v_i to v_j. Also by the first remark, vertex transitions appear as a whole, thus S' will consist of all the vertex subsequences with edge transitions in between. This is an ordering of all vertices connected by edges, and thus is a Hamiltonian path. Minimum total cost is $3|V| + (M+1)|E|$, which is achieved only when the solution defines a Hamiltonian path. So we conclude that G has a Hamiltonian path if and only if the cost of the compacted sequence is $3|V|+(M+1)|E|$. The cost of a solution for the compaction problem can be verified in polynomial time, thus the sequence compaction problem is NP-complete. □

2.2 Graph Model for Sequence Compaction

This section describes our graph model for the sequence compaction problem, where a heaviest-weighted trail describes an optimally compacted sequence. In this model, distinct vectors and transitions are represented by vertices and multiple weighted directed edges, respectively. Since edges of the graph correspond to transitions of the sequence, one more occurrence of an edge in the trail implies one more occurrence of this transition in the compacted sequence. So, we assign edge weights as the decrease in $\mathcal{C}(S, S', c)$ when the corresponding transition appears in the compacted sequence. Formally, each transition t_i in S, is represented by several edges $e_{i1}, e_{i2}, \ldots, e_{ij}, \ldots$ with $w(e_{i1}) \geq w(e_{i2}) \geq \ldots \geq w(e_{ij}) \geq \ldots$. Thus the weight of the jth edge can be set as the change in cost, if transition t_i is added once more to S'.

$$w(e_{ij}) = \frac{|S(t_i) - c(j-1)| - |S(t_i) - cj|}{S(t_i)} = \begin{cases} \frac{c}{S(t_i)} & \text{if } S(t_i) \geq cj \\ \frac{-c}{S(t_i)} & \text{if } S(t_i) \leq c(j-1) \\ \frac{2(S(t) \bmod c) - c}{S(t_i)} & \text{otherwise} \end{cases}$$

A graph $G = (V, E)$ that represents the sequence compaction problem for input sequence $S = \langle s_1, \ldots, s_m \rangle$ and compaction factor c, has a vertex v_i for each input vector s_i in S. Each transition $t = (s_i, s_j)$ in S is represented by multiple copies of edge (v_i, v_j): *i)* $\lfloor \frac{S(t)}{c} \rfloor$ copies with weight $\frac{c}{S(t)}$ *ii)* one copy with weight $\frac{2(S(t) \bmod c) - c}{S(t)}$, if $\lfloor \frac{S(t)}{c} \rfloor \neq \frac{S(t)}{c}$ *iii)* sufficiently many copies with weight $\frac{-c}{S(t)}$. Edges generated by the first rule correspond to transitions that will always reduce $\mathcal{C}(S, S', c)$, whereas edges generated by the last rule increase the cost to penalize overestimation. The second rule corresponds to approximating $\frac{S(t)}{c}$ with either $\lfloor \frac{S(t)}{c} \rfloor$ or $\lceil \frac{S(t)}{c} \rceil$. Parallel edges with identical weights can be packed into one edge with *multiplicity*. Notice that there are at most three different weights for each edge. Thus the size of the graph is determined by the number of distinct input vectors (the number of vertices) and the number of distinct transitions (the number of edges). By this edge-weight assignment scheme, a heaviest-weighted trail defines an optimally compacted sequence.

Notice that our model does not rely on the particular cost function we are using. Our model is still valid when a different cost function is employed. All we need to do is assign edge weights as the difference in the total cost function when the corresponding edge appears once more in the solution.

2.2.1 *Finding a Heaviest-Weighted Trail.* Our heuristic has three steps. First we detect all positive weight cycles by the Bellman-Ford algorithm [Cormen et al. 1990] and remove them. Then we find a heaviest-weighted trail in the reduced graph by using the the Bellman-Ford algorithm, and finally we add the positive weight cycles back to the trail. If a vertex v in the cycle is on the trail, then the cycle can be inserted to the trail by opening it at vertex v to make a trail with v at both ends, and replacing vertex v of the original trail with this trail.

2.2.2 *Constructing Multiple Sequences.* More accurate estimations can be achieved with even shorter sequences by generating multiple sequences. A high compaction factor can be used for a shorter sequence; then accuracy can be restored by a second sequence with a lower compaction factor that covers transitions underestimated by the first one. Consider a sequence where $s_1 = \langle ABCDA \rangle$ is repeated four times,

Table I. Misprediction percentages for SPICE simulations

Circuit	$c=3$		$c=5$ Low		$c=5$ Medium		$c=5$ High		$c=10$	
	MM	HWT	MM	HWT	MM	HWT	MM	HWT	MM	HWT
i3	2.4	0.6	2.1	0.7	2.2	0.7	2.5	0.8	1.7	1.2
cordic	4.3	1.3	4.6	2.8	4.8	3.1	5.2	3.1	4.8	3.2
C432	3.0	0.8	2.7	0.5	2.6	0.7	2.9	0.7	3.1	1.1
C880	5.1	0.8	5.1	1.8	5.2	1.9	5.6	2.1	5.8	2.2
C1355	8.6	1.5	8.2	2.1	9.3	2.4	10.7	2.5	9.4	2.7
C1908	3.3	1.6	3.8	2.5	4.2	2.5	4.7	2.6	5.3	2.9
C3540	3.8	1.1	4.2	1.6	4.6	1.6	5.0	1.8	5.1	2.0
C6288	6.5	1.4	8.2	2.5	8.3	2.6	8.8	2.8	8.6	3.2
Average	4.6	1.1	4.9	1.8	5.2	1.9	5.7	2.1	5.5	2.3

whereas $s_2 = \langle ACBEA \rangle$ appears only twice. If we generate a single sequence for $c=2$, then the compact sequence will have two copies of s_1 and one copy of s_2. However, we can generate two sequences $\langle ABCDA \rangle$ and $\langle ACBEA \rangle$ with $c{=}4$ and $c=2$, respectively, for the same accuracy with shorter sequences. So, we define the problem as: *Given a sequence S, and compaction factors $c_1 \geq \ldots \geq c_k$, construct sequences $S'_1, \ldots, S'_k$ to minimize $\sum_{t \in T(S)} \frac{|S(t) - \sum_{i=1}^{k} c_i S'_i(t)|}{S(t)}$.* Notice that compacted sequences preserve transition frequencies collectively, not individually. To generate such sequences, we start by constructing S'_1 for c_1, and update edge weights to reflect what is already covered by S_1 before constructing S'_2, edge weights considering S'_1. In general, while constructing the graph for S_i, an edge weight is computed as the change in cost when this transition occurs once more in S_i, and we take into account what is already covered by preceding sequences. Let $P_i(t) = \sum_{j=1}^{i-1} S_j(t) c_j$ denote how many times t is covered in the first $i-1$ compacted sequences. Weight of the jth edge $w(e^i_j)$ for transition t to construct S_i is set as $w(e^i_j) = \frac{|S(t) - P_i(t) - c_i(j-1)| - |S(t) - P_i(t) - c_i j|}{S(t)}$. Corresponding sequence S_i can be constructed by finding a heaviest-weighted trail.

3. EXPERIMENTAL RESULTS

Three metrics are used in the experiments: accuracy of estimations for circuit simulations, reliability: preservation of transition frequencies, and closeness to optimality. For circuit simulations, we worked with SPICE on eight MCNC91 circuits, and three compaction factors: 3, 5, 10. We measured power consumption of circuits for 6 biased sequences of length 2000. The sequences were compacted using Markov model (MM) [Marculescu et al. 1997b] and the proposed heaviest-weighted trail method (HWT). Table I presents average accuracies in percentages. We see that HWT can predict the power consumption very accurately. The table also breaks down the experiments according to bit-switching activity for $c{=}5$. Tracing transition frequencies is more crucial for higher bit-switching activity, and HWT retains its accuracy as the switching activity increases.

Compaction methods are proposed to avoid long simulations; thus accuracy cannot be verified. A reliable solution estimates each transition accurately, which implies accuracy in estimations. We compared the reliabilities of HWT and MM solutions for 750 sequences of length 20,000 and seven compaction factors. As seen

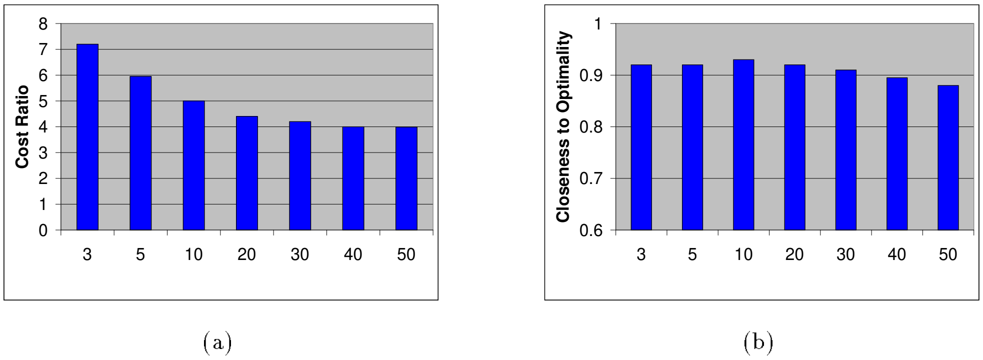

Fig. 1. (a) Ratios of costs of solutions with HWT and MM; (b) Closeness to optimality for HWT.

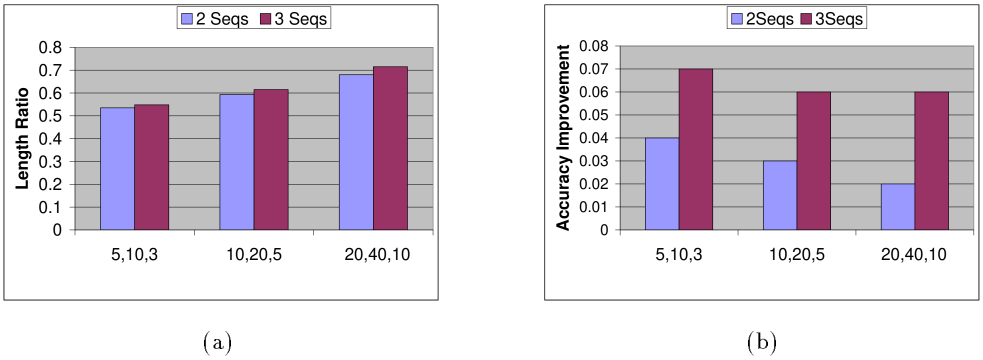

Fig. 2. Solution qualities for multiple sequences: (a) length; (b) accuracy.

in Figure 1(a), there is a drastic difference in cost values, thus preserving transition frequencies between the two methods. For $c = 3$, cost of an MM solution is more than seven times larger than that of HWT. The ratio decreases as the compaction factor increases, and performances virtually get closer. However, this should be attributed to the increasing cost of an optimal solution.

We also investigated closeness to optimality. As a bound on the optimal solution value, we assumed a hypothetical sequence that estimates each transition optimally, which also corresponds to a trail that covers all and only positive weight edges. Figure 1(b) presents results for compacting 750 sequences of length 20,000. In this figure, 1 is the bound on the optimal solution value. As we see, the solutions are within 10% of upper bounds.

Figure 2 compares generating two and three sequences with a single sequence on compaction factor sets. The first (second) column in each set corresponds to generating two (three) sequences with the first two (three) compaction factors displayed below the columns. Single sequence solutions use the first number as the compaction factor. Figures. 2(a) and (b) compare sequence lengths and accuracies of the two methods, respectively. The first set shows that it is possible to get a more accurate solution with almost half the total sequence length. The third sequence improves accuracy but slightly increases total sequence length. Finally, Table II

Table II. SPICE simulations for MS and HWT

Circuit	$c = 3,5,2$		$c = 5,8,3$		$c = 10,15,7$	
	HWT	MS	HWT	MS	HWT	MS
i3	0.6	0.4	0.7	0.5	1.2	0.5
cordic	1.3	1.0	3.0	1.6	3.2	2.1
C432	0.8	0.6	0.6	0.5	1.1	0.7
C880	0.8	0.5	1.9	1.2	2.2	1.1
C1355	1.5	1.1	2.3	1.6	2.7	1.7
C1908	1.6	0.9	2.5	1.4	2.9	1.7
C3540	1.1	0.8	1.7	1.1	2.0	1.2
C6288	1.4	1.0	2.6	1.8	3.2	2.3
Averages	1.1	0.8	1.9	1.2	2.3	1.4

presents average error in SPICE simulations for six sequences of length 2000. MS corresponds to generating three sequences with compaction factors listed, and HWT corresponds to generating one sequence with the first compaction factor at the top of the column. Results show that accuracies can be improved by generating multiple sequences, and justify generating multiple sequences as a valuable alternative for "harder" sequences.

4. CONCLUSION

We addressed the sequence compaction problem for efficient and accurate power estimation. We proved that transforming a sequence into a smaller one with minimum deviation in transition frequencies is NP-complete. We also proposed a graph model to reduce the compaction problem to that of finding a heaviest-weighted trail and a heuristic to find such trails. Generating multiple compact sequences with different compaction factors is also discussed. Proposed methods were applied to MCNC 91 benchmark circuits, using SPICE for simulations. Results showed that our methods can significantly reduce simulation times with very high accuracy.

REFERENCES

Cormen, T., Leiserson, C., and Rivest, R. 1990. *Introduction to Algorithms*. MIT Press.

Huang, S., Chen, K., Cheng, K., and Lee, T. 1996. Compact vector generation for accurate power simulation. In *Proc. 33rd Design Auto. Conf.* 161–164.

Marculescu, R., Marculescu, D., and Pedram, M. 1997a. Hierarchical sequence compaction for power estimation. In *Proc. 34th Design Auto. Conf.* 570–575.

Marculescu, R., Marculescu, D., and Pedram, M. 1997b. Vector compaction using dynamic markov models. *IEICE Trans. Fundamentals of Electronics Communications and Computational Sciences E80-A,* 10, 1924–1933.

Marculescu, R., Marculescu, D., and Pedram, M. 1999. Sequence compaction for power estimation: Theory and practice. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems 18,* 7, 973–993.

Pedram, M. 1999. *Power simulation and estimation in VLSI circuits.* The CRC Press and the IEEE Press.

Pinar, A. and Liu, C. 1998. Power invariant vector sequence compaction. In *Proc. 1998 Int. Conf. Computer-Aided Design*. 473–476.

Pinar, A. and Liu, C. 2002. Sequence compaction to preserve transition frequencies. Tech. Rep. LBNL-51864, Lawrence Berkeley National Laboratory, Berkeley, CA.

Tsui, C., Marculescu, R., Marculescu, D., , and Pedram, M. 1996. Improving efficiency of power simulators by input vector compaction. In *Proc. 33rd Design Auto. Conf.* 165–168.

Received September 2000; revised January 2002; accepted November 2002