International Symposium on Low Power Electronics and Design, 1997, pp. 178-183

K2: An Estimator for Peak Sustainable Power of VLSI Circuits

Michael S. Hsiao†, Elizabeth M. Rudnick††, and Janak H. Patel††
†Department of Electrical and Computer Engineering
Rutgers University, Piscataway, NJ
††Center for Reliable and High-Performance Computing
University of Illinois, Urbana, IL

Abstract

New measures of peak power in the context of sequential circuits are proposed. This paper presents an automatic procedure to obtain very good lower bounds on these measures as well as the actual input vectors that attain such bounds. The initial state of the circuit is an important factor in determining the amount of switching activity in sequential circuits and is taken into account. A peak power estimator tool *K2* was developed using genetic techniques. Experiments show that vector sequences generated by *K2* give much more accurate estimates for peak power dissipation than the estimates made from randomly generated sequences.

I Introduction

One of the major design constraints for VLSI circuits in recent years has been power dissipation because of the continuing increase in chip density. Excessive power dissipation can cause overheating, which reduces the life-time of the chip and degrades the circuit's performance. As pointed out in [1], large instantaneous power dissipation can cause overheating (local hot-spots), and the failure rate for components roughly doubles for every $10^{\circ}C$ increase in operating temperature. The power dissipated in CMOS logic circuits is a complex function of the gate delays, clock frequency, process parameters, circuit topology and structure, and the input vectors applied. Once the processing and structural parameters have been fixed, the measure of power dissipation is dominated by the switching activity (toggle counts) of the circuit. Peak power measures can serve as guidelines for boundaries and limits of a circuit.

Much effort has been invested in estimating average power dissipation [2-7]. Most commonly, the average power is estimated from signal switching probabilities. Average power dissipation, however, is insufficient in providing accurate boundaries for estimating limits of the design. The problem of estimating the worst-case power dissipation in CMOS *combinational* circuits has been addressed in [8]. The worst-case power is transformed into a weighted max-satisfiability problem on a set of multi-output boolean functions, obtained from the logic description of the circuit. Either a disjoint cover enumeration or branch-and-bound algorithm is used to solve the complex NP-complete max-satisfiability problem. The largest circuit reported in [8] has only 733 gates, and several hours of CPU time are required for computation. Peak current estimation for combinational circuits was addressed in [9, 10]. The authors' approach was to find the time window during which a gate in the circuit could switch. Maximum power cycles were computed using symbolic transition counts in [11]; the state transition diagram (STG) was used to find the maximal average cycle in the graph, where the edges in the STG indicate the power dissipation between two adjacent states. In a small circuit, s208 for instance, the dual graph necessary to compute the power dissipation contains 71 million edges [11]; the largest circuit reported has less than 500 gates, and handling large circuits with large numbers of flip-flops is impractical. Finally, peak power estimation was computed for sequential circuits using a test generation based technique in [12]. Attempts were made to create toggles in the circuit for gates with the greatest numbers of fanouts. The estimates were based on a zero-delay fault-simulation model and did not extend to cover peak sustainable power.

The focus of this work is to propose new measures of peak power in the context of sequential circuits and develop an automatic procedure to obtain very good lower bounds on these measures, as well as providing the actual input vectors that attain such bounds. The peak powers that we will define are peak average powers, where the average is computed over different time periods. The three measures that we will use in this paper are: *Peak Single-Cycle Power*, *Peak n-Cycle Power*, and *Peak Sustainable Power*, covering time durations of one clock cycle, several consecutive clock cycles, and an infinite number of cycles, respectively. The unit of power used throughout the paper is energy per clock cycle and will simply be referred to as power.

Peak Single-Cycle Power is the maximum total power consumed during one clock cycle.

Peak n-Cycle Power is the maximum average power of a contiguous sequence of n clock cycles, assuming that the initial state of the machine is fully controllable.

Peak Sustainable Power is the maximum average power that can be sustained indefinitely over many clock cycles.

The definitions are illustrated in Figure 1, where a sequential circuit is shown unrolled into several clock cycles, commonly known as an iterative logic array (ILA) representation of the sequential circuit. In a typical sequential circuit, the switching activity is largely controlled by the state vectors and less influenced by input vectors, because the number of flip-flops far outweighs the number of primary inputs. For this reason, it is

*This research was conducted at the University of Illinois and was supported in part by the Semiconductor Research Corporation under contract SRC 96-DP-109, in part by DARPA under contract DABT63-95-C-0069, and by Hewlett-Packard under an equipment grant.

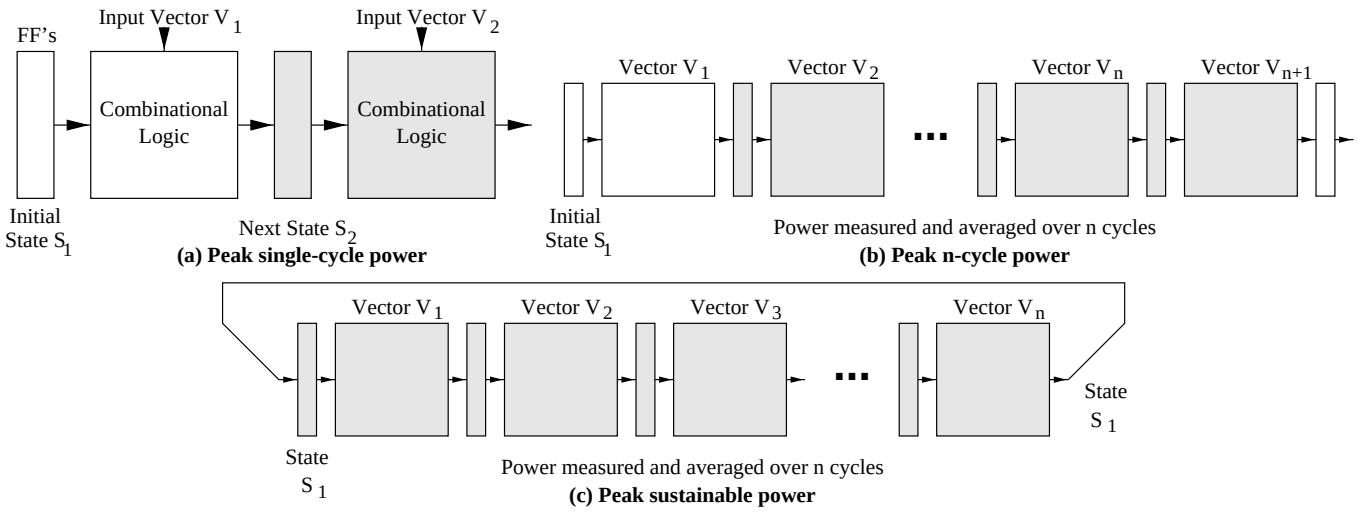

Figure 1: Definitions of Peak Power Measures during shaded cycles.

important to understand the differences in the three measures defined above.

As illustrated in Figure 1(a), the peak single-cycle power is controlled by initial state S_1 and input vectors V_1 and V_2. The state S_1 and input vector V_1 initialize all circuit nodes and determine the next state S_2. Then vector V_2 and state S_2 switch some of the gates, which accounts for the power dissipation. We will obtain a three-tuple (S_1, V_1, V_2) that tries to maximize this power. Since the procedure to obtain this three-tuple is imperfect, the result will be a lower bound on this measure. What is the utility of this measure? In fully-scanned circuits, the state S_1 can be initialized to any arbitrary value, and therefore, this bound is attainable in practice. However, in cases where the initial state is not fully controllable, we can only speculate that during the operation of the circuit, the machine may reach state S_1, and only then can we be assured that the bound is attainable. If the computed state S_1 is not a valid state (i.e., not reachable in the normal operation), then we may not be able to reach that power level. If we put the restriction that the state S_1 be a valid state, the peak single cycle power obtained could be lower than the value obtained without any restriction. In the method proposed later in the paper, we can obtain peak power with or without the restriction of reachability of the state S_1.

Peak n-Cycle power is illustrated in Figure 1(b). We will search for an $(n+2)$-tuple (S_1, V_1, ..., V_n, V_{n+1}) that maximizes the power over n cycles. Sequential circuits place considerable constraints on the sequence of consecutive states that can be traversed. Therefore, this peak power will always be less than or equal to the peak single-cycle power. Utility of this measure is in thermal management of the package. Peak single-cycle power is close to the instantaneous peak power, which is mostly a transient event. However, for a reasonable size n, the peak n-cycle power could represent a practical worst case for heat dissipation. We could also restrict the initial state S_1 to a known valid state, in which case the peak could be lower than the peak obtained without any restriction on S_1.

Peak sustainable power is illustrated in Figure 1(c). The state S_1 is repeated at the end of the input sequence. As a result, the power level can be maintained by applying this input sequence again and again. Clearly, this peak power measure is very important for thermal management of a chip.

In all cases, the power dissipated in the combinational portion of the sequential circuit can be computed as

$$P = \frac{V_{dd}^2}{2 \times clock\ period} \times \sum_{for\ all\ gates\ g} [toggle(g) \times C(g)],$$

where $toggle(g)$ is the number of switches (0 to 1 or vice versa) for gate g in a cycle period, and $C(g)$ represents the output capacitance of gate g. Since the output capacitance can be approximated by a constant times the number of fanouts of the gate, the power expression can now be re-written as

$$P = \frac{V_{dd}^2 \times C_{load}^1}{2 \times clock\ period} \times \sum_{for\ all\ gates\ g} [toggle(g) \times fanout(g)],$$

where C_{load}^1 is a unit-load capacitance per node. Switching rate per node is compared across different circuits, so we report switching frequency per node instead of total power; the switching frequency is computed as $SF = Q/(number\ of\ capacitive\ nodes)$, where $Q = \sum[toggle(g) \times fanout(g)]$, over all gates g.

Because peak power estimation involves the maximization of a switching-activity function $toggle(g) \times fanout(g)$, optimization algorithms are needed. Genetic algorithms (GA's) are chosen as the optimization tool for this problem, since GA's have been shown to be very successful in a wide range of applications where optimization is involved. Specifically, GA's have been quite effective in the area of automatic test pattern generation for large sequential circuits [13-16]. GA's are used in this work to find the vector sequences which most accurately estimate the peak single-cycle, n-cycle, and sustainable power dissipations. The estimates obtained are compared with the estimates from randomly-generated sequences. The GA-based estimates will be shown to achieve much tighter lower-bounds on the peaks.

The remainder of the paper presents the design and analysis of the peak estimation tool **K2**. Section II describes briefly the genetic algorithms used in **K2**; Section III explains the details of finding the peak single-cycle and n-cycle power dissipation; Section IV discusses the issues of estimating peak sustainable power for the circuit; experimental data from running **K2** on several benchmark circuits are discussed in Section V, and Section VI concludes the paper.

II Genetic Algorithms

The GA framework used in the implementation of **K2** is similar to the simple GA described in [17]. The GA contains a population of *strings*, also called *chromosomes* or *individuals*, in which each individual represents a sequence of vectors preceeded by the initial state. Peak n-cycle power estimation requires a search for the $(n+2)$-tuple (S_1, V_1, ..., V_n, V_{n+1}) that maximizes power dissipation. This $(n+2)$-tuple is encoded as a single binary string. The population size used is a function of the string length, which depends on the number of primary inputs, the number of flip-flops, and the vector sequence length n. Larger populations are needed to accommodate longer sequences in order to maintain diversity. The population size is set equal to $32 \times \sqrt{sequence\ length}$ when the number of primary inputs is less than 16 and $128 \times \sqrt{sequence\ length}$ when the number of primary inputs is greater than or equal to 16.

Each individual has an associated *fitness*, which measures the quality of the individual in terms of switching activity. Because each fanout on a gate contributes to the output capacitance of that gate, the number of gate fanouts is taken into account in the fitness function: $fitness = \sum toggle\ count(g) \times fanout(g))$.

The population is first initialized with random strings. A unit-delay logic simulator is then used to compute the fitness of each individual. The evolutionary processes of *selection*, *crossover*, and *mutation* are used to generate an entirely new population from the existing population. Evolution from one generation to the next is continued until a maximum number of generations is reached. In this work, 32 generations are allowed. To generate a new population from the existing one, two individuals are selected, the two individuals are crossed to create two entirely new individuals, and each character in a new string is mutated with some small mutation probability. A mutation probability of 0.01 is used in this work, and since a binary coding is used, mutation is done by simply flipping the bit. The two new individuals are then placed in the new population, and this process continues until the new generation is entirely filled. At this point, the previous generation can be discarded. In our work, we use tournament selection without replacement and uniform crossover. In *tournament selection without replacement,* two individuals are randomly chosen and removed from the population, and the best is selected; the two individuals are not replaced into the original population until all other individuals have also been removed. Thus, it takes two passes through the parent population to completely fill the new population. In *uniform crossover*, bits from the two parents are swapped with probability 0.5 at each string position in generating the two offspring. A crossover probability of 1 is used; i.e., the two parents are always crossed in generating the two offspring. Because selection is biased toward more highly fit individuals, the average fitness is expected to increase from one generation to the next. However, the best individual may appear in any generation, so we save the best individual found.

Since the majority of time spent by the GA is in the fitness evaluation, parallelism among the individuals can be exploited. Parallel-pattern simulation [18] is used to speed up the process; thus, 32 candidate sequences from the population are simulated simultaneously, with values bit-packed into 32-bit words.

III Peak Single-Cycle and N-Cycle Powers

We estimate the power dissipation in CMOS circuits by measuring the amount of gate switching activity; static power dissipation is neglected. Two components make up the switching activity: zero-delay activity and spurious activity such as glitches and hazards. This work considers the unit-delay model for simulation. However, the methodology is independent of the underlying simulator, and different timing simulators could be used in place of the unit-delay simulator. The output capacitance (measured by the number of fanouts on a given gate) is accounted for by the GA in the fitness function. Here the assumption is made that gate capacitance is proportional to the number of fanouts. However, assigned output capacitances for the gate output nodes can be handled by our optimization technique as well.

Peak single-cycle switching activity occurs when the greatest number of nodes are toggled between two consecutive vectors. For combinational circuits, the task is to search for a pair of two vectors (V_1, V_2) that generates the most gate transitions. For sequential circuits, on the other hand, the activity depends on the initial state as well as the primary input vectors. For the pair of vectors to be useful, the initial state needs to be reachable. The estimate for peak power dissipation can be used as a lower-bound for worst-case power dissipation in the circuit in any given time frame.

Peak n-cycle power is a measure of the maximum average switching activity over a contiguous sequence of n vectors. This measure serves as an upper-bound to *peak sustainable power*, and it is considered only for sequential circuits. The initial state of the sequence must also be reachable. The n-cycle power dissipation varies with the sequence length n. When n is equal to 1, the power dissipation is the same as the peak single-cycle power dissipation, and as n increases, the average peak power is expected to decrease if the peak single-cycle power dissipation cannot be sustained over the n vectors in the sequence. Figure 2 illustrates a typical curve for the peak power dissipation as a function of vector sequence length. The peak levels off as the sequence length approaches infinity or earlier when a loop is found in the state transitions.

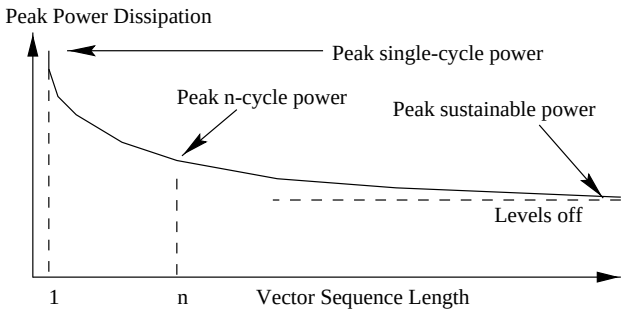

Figure 2: Lower Bound of Peak Power Dissipation.

In order to take the initial state into consideration during power estimation for sequential circuits, the *state* portion of the individuals in the GA may be seeded using a previously computed set of reachable states of the circuit, S_{reach} (this set may not be complete). After several generations of the evolutionary processes, the *optimized* individuals may contain states that are not in S_{reach}. At this point, attempts can be made to prove the reachability of the required state with the use of a sequential test generation state justification procedure. State justification, however, is a very complex problem involving a large number of backtracks [19, 20]. In order to reduce the ex-

ecution time, an alternative to state justification is taken. A set of states, S_{sim}, is formed by selecting states from S_{reach} that are similar to the required target state. A state S_i is similar to another state S_j if the *Hamming distance* between their encodings is small. Hamming distance is the number of bits having different values. The best vector-sequence generated by the GA is simulated from every state in S_{sim}, and the maximum power obtained from the set of states is taken as the peak power. Again, parallelism is exploited among the different starting states during simulation. The state from which the most switching activity is obtained is selected to be the initial state for the sequence.

IV Peak Sustainable Power

The peak single-cycle and n-cycle power estimates discussed in the previous section only last as long as the length of the sequence. Another valuable and useful measure is the peak *sustainable* power. An interesting question, therefore, is how to estimate the peak sustainable power, and whether a long sequence of vectors can be derived such that a very high power dissipation is generated and sustained.

The method involving symbolic transition counts to compute maximum power cycles in [11] is impractical due to the huge sizes of STG's and binary decision diagrams for large circuits. Our approach, on the other hand, avoids the STG and symbolic techniques entirely.

From the curve of the lower-bound of peak power dissipation illustrated in Figure 2, the lower-bound for maximal power dissipation reaches a *steady state* when the sequence length goes to *infinity*. In this work, the term **peak sustainable power dissipation** denotes the steady-state value. Consider the cases illustrated in Figure 3, where the state-vector three tuple (A, V_i, V_j) generates the peak single-cycle power dissipation for the chip; in addition, V_i and V_j are associated with a loop back to state A in the state machine of the circuit. If the initial vectors of a sequence T_{sus} take the circuit to state A, and vectors V_i and V_j are repeated for the remainder of T_{sus}, the peak power dissipation can be sustained for every time-frame after state A has been reached, and sequence T_{sus} becomes the peak-power-sustaining sequence. Unfortunately, it is nontrivial to find loops, especially when we are working without a state diagram. This problem is complicated further in that we also want to maximize the power dissipation in loops. We could approach this problem in the following way. First, find a peak n-cycle power sequence. Then, try to close the loop with as few additional state transitions as possible. The problem with this approach is that closing the loop is a hit-or-miss proposition. Moreover, the additional transitions will reduce the peak power, since their selection is primarily based on the objective to close the loop, not to maximize power. Another approach is to derive a peak n-cycle power sequence starting from an initial state S_{easy}, which is easy to reach from any state, then close the loop with only a few additional transitions. We took this approach to an extreme, starting with the entirely *don't care* state. Since this state is a superset of any state, it can be reached in just one transition from any state. We derive a peak n-cycle power sequence starting from the all-unknown state; the sequence is always a loop because the final state of the sequence is covered by the initial state. Because the final state is a fully-specified state, the derived sequence is also a synchronizing sequence. In fact, any synchronizing sequence is a loop, as shown in Figure 4, when the initial state is set equal to the final state of a synchronizing sequence. Power dissipation is measured inside the loop, starting from state S_n, when the flip-flops are initialized. This approach restricts the search of peak power loops to a subset of all loops and thus may not be a very tight lower bound. However, our experiments will show that this approach still yields peaks higher than extensive random search.

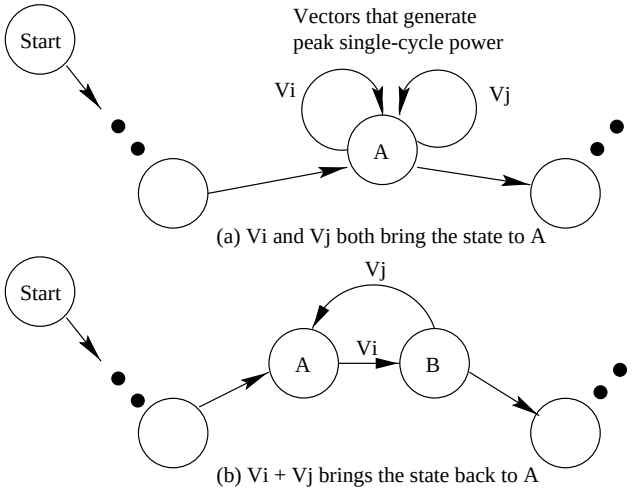

Figure 3: Ideal Cases Involving Self-Loops.

V Experimental Results

The power estimation algorithms were implemented in a tool called **K2** using the C++ language; ISCAS85 combinational benchmark circuits, ISCAS89 sequential benchmark circuits [21], and several synthesized circuits [16] were used to evaluate the speed and accuracy of the peak power estimator **K2**. Results were also reported for proc16, which is a 16-bit microprocessor with 56 flip-flops and 8905 gates. All computations were performed on an HP 9000 J200 with 256 MB RAM.

Before we begin to discuss the results, we will show that the estimates made by our GA-based technique are indeed good estimates of peak power. In other words, how close are our estimates to the theoretical peaks? To partially answer this question, we performed extensive simulations. Millions of random vector pairs were simulated for seven small sequential circuits under the unit-delay model, and results are shown in Table 1. All peak powers are expressed in terms of the peak switching frequency (PSF) averaged per node in the circuit. The first 4 columns (PSF and Time) for random sequences are taken at 2 and 100 million random vector pairs. The number of vector pairs required for exhaustive simulation is similar. For instance, in circuit s400 which has 21 flip-flops and 3 primary inputs, the

Table 1: **K2** vs. Random on Peak Single-Cycle Power

Ckt	**Random**				**GA-based**	
	2 million		**100 million**		**(∼ 64,000)**	
	PSF	**Time**	**PSF**	**Time**	**PSF**	**Time**
s298	1.015	6.5 m	1.015	5.39 h	**1.015**	5.32 s
s382	1.045	8.8 m	1.063	7.29 h	**1.081**	10.3 s
s400	1.049	9.1 m	1.072	7.58 h	**1.106**	13.4 s
s444	1.069	10.9 m	1.113	9.01 h	**1.150**	12.7 s
s526	0.903	10.5 m	0.922	8.65 h	**0.930**	11.3 s
s641	2.567	27.3 m	2.749	22.3 h	**2.869**	56.2 s
s713	2.649	30.3 m	2.749	24.8 h	**2.815**	50.2 s

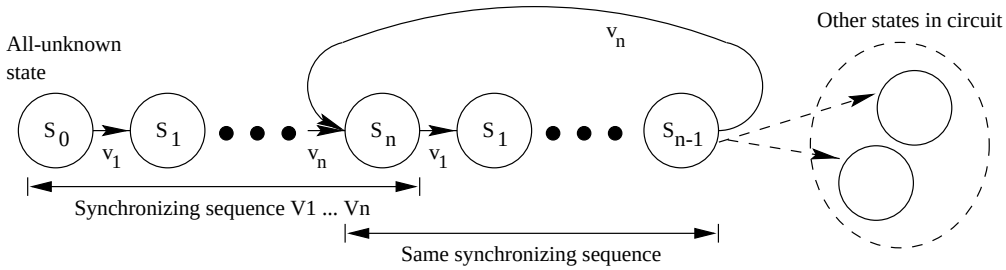

Figure 4: Power Estimation with Synchronizing Loops.

total number of exhaustive simulations for the three-tuple (S_1, V_1, V_2) would be $2^{(21+3+3)} = 2^{27} \simeq 134$ *million*. The rightmost columns show the results obtained by **K2**, which required approximately 64,000 simulations. All the estimates were made without performing reachability analysis. For the five circuits s298 to s526, the 100 million simulation results are similar to **K2** estimates, however, at the cost of many additional hours of computation. For the last two circuits, s641 and s713, the results from simulation of 100 million random sequences (over twenty-two hours of execution) still lag 4.4% behind **K2** estimates, which took less than one minute each. This exercise shows that estimates from **K2** give tight lower-bounds indeed.

Results for peak single-cycle power dissipation in combinational circuits will be discussed first. The estimates are displayed in Table 2. For each circuit, the total number of capacitive nodes in the circuit (computed as the total number of gate inputs in the circuit) is given, followed by the peak power dissipation expressed in terms of the peak switching frequency averaged per node, and finally, the execution time. We made the assumption that all nodes have identical input capacitance (fanouts are accounted for by the number of stems from that node); however, assigned capacitances for the circuit nodes can be handled by our optimization technique as well. Estimates made by **K2** are significantly higher than the estimates made from random vector sequences for most circuits. 49.2% improvement in the estimate is made for circuit c2670; an average of 27.4% improvement is achieved for the combinational circuits, as shown at the bottom of the table. The numbers of vector pairs used are identical for random simulation and **K2**. Thus, execution times are comparable between **K2** and the random approach for all circuits; **K2** sometimes takes slightly longer time to finish, due to the execution of genetic operations from one generation to the next.

Table 2: Peak Power for Combinational Circuits

Circuit	# Cap Nodes	Random		K2	
		PSF	Time	PSF	Time
c432	343	0.872	6.2 s	**1.175**	7.2 s
c499	440	**0.995**	6.2 s	**0.995**	7.2 s
c880	755	0.656	12.2 s	**0.823**	13.2 s
c1355	1096	0.830	13.3 s	**0.883**	13.3 s
c1908	1523	1.333	32.3 s	**1.838**	34.3 s
c2670	2216	0.995	88.9 s	**1.464**	89.9 s
c3540	2961	1.301	77.1 s	**1.460**	74.1 s
c5315	4509	1.241	164 s	**1.714**	170 s
c6288	4832	6.317	594 s	**7.359**	617 s
c7552	6252	1.518	314 s	**2.020**	311 s
% Impr				27.4%	

Maximal peak power highlighted in **bold** for each circuit

Results for sequential circuits are shown in Table 3. The reachability analysis for sequential circuits was performed for the initial state, and the corresponding power dissipations are reported for estimates computed after the reachability analysis. The time required to perform reachability analysis was minimal, so it is not included. The set of reachable states was computed prior to power estimation using random simulation, and the execution times for reachable state computation are not included in the tables. The set of reachable states may be derived by several different methods: from the state table, random vector simulation, an automatic test generator, or a GA-based approach. Sequence lengths of 10 are used for n-cycle and sustainable power estimates.

Table 3: Peak Power for Sequential Circuits

Circuit	# Cap Nodes	Single-cycle		10-cycle	Sustainable
		PSF	Time	PSF	PSF
s298	264	0.856	5.3 s	0.483	0.413
s344	295	0.810	12.7 s	0.683	0.683
s382	333	0.931	10.3 s	0.368	0.262
s400	349	0.908	13.4 s	0.375	0.252
s444	379	0.953	12.7 s	0.372	0.281
s526	472	0.642	11.3 s	0.321	0.241
s641	582	1.701	56.2 s	0.902	0.902
s713	633	1.599	50.2 s	0.964	0.964
s820	781	0.855	28.1 s	0.682	0.628
s832	793	0.874	29.3 s	0.685	0.647
s1196	1041	1.022	44.1 s	0.775	0.775
s1238	1073	1.043	19.8 s	0.796	0.796
s1423	1243	1.459	54.0 s	0.867	0.604
s1488	1412	1.176	78.4 s	0.778	0.663
s1494	1418	1.171	79.0 s	0.790	0.657
s5378	4440	1.011	563 s	0.648	0.465
s35932	30317	1.123	23200 s	1.100	1.107
am2910	1998	3.224	344 s	1.715	1.295
mult16	1323	1.238	156 s	0.941	0.787
div16	1760	3.774	521 s	1.996	1.781
proc16	9094	1.572	2087 s	0.490	0.203
% Impr		8.9%		18.2%	23.8%

In many practical circuits, the portion of states reachable out of the 2^N possible states becomes smaller when the number of flip-flops, N, increases. Thus, the optimized initial state is frequently unreachable for circuits with many flip-flops, resulting in a lower peak power measure after the reachability analysis. Reachability analysis is not needed for peak sustainable power because synchronizing sequences start the circuit in the *don't-care* state. The **K2** estimates provide *tighter* bounds for all circuits when compared with the estimates from the randomly generated sequences. Averages of 8.9%, 18.2%, and 23.8% im-

provements were observed for peak single-cycle, 10-cycle, and sustainable powers, respectively, over the randomly-generated vector sequences. The number of vector pairs used in random simulation and **K2** are also identical. The execution times needed for 10-cycle and sustainable power are not reported, but they are always less than 10 times that of the single-cycle power because the amount of switching activity incurred is not ten times as much.

Figure 5 illustrates graphically the relationships among the estimates of peak single-cycle, 10-cycle, and sustainable power. It is intuitive that neither peak 10-cycle nor sustainable power

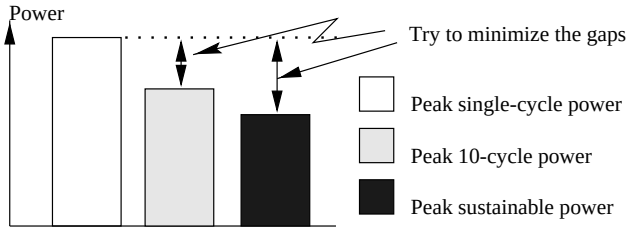

Figure 5: Relationship Among Peak Single-Cycle, N-Cycle, and Sustainable Powers.

estimates should be as high as the peak single-cycle estimate, unless the peak single-cycle power can be sustained in a loop fashion as described in an earlier section. Furthermore, the peak sustainable power estimates are expected to be lower than the peak 10-cycle power because the sequences that produce peak 10-cycle power are not restricted to be loops. In many circuits, significant gaps exist between the peak single-cycle and 10-cycle power estimates, indicating that the peak single-cycle power dissipations are difficult to sustain for these circuits. However, among these circuits, some of them, including s344, s641, s832, s1238, s35932, am2910, and div16, have peak sustainable power estimates very near their peak 10-cycle power lower-bounds; this suggests that the lower-bound peak sustainable power curves (i.e., Figure 2) are likely to level off near the sequence length $n = 10$ for these circuits. For the remaining three circuits, the lower-bound curves level off at longer sequence lengths.

VI Conclusions

A GA-based power estimation framework, **K2**, was presented. Estimates for peak single-cycle, n-cycle, and sustainable power dissipation are computed for the unit-delay model; the role of the initial state in sequential circuits has also been taken into account for a more accurate measure. **K2** provides much tighter bounds when compared to the estimates made from randomly-generated sequences. The average improvements in the estimates between the two approaches were 27.4% for combinational circuits and up to 23.8% for sequential circuits. In addition, the execution times of **K2** were orders of magnitude lower than those for random-based estimates, if they were to achieve similar tightness of lower bounds.

References

[1] C. Small, "Shrinking devices put the squeeze on system packaging," *EDN.*, pp. 41-46, Feb. 1994.

[2] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," *Proc. Design Automation Conf.*, pp. 253-259, 1992.

[3] F. N. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Trans. on VLSI Systems*, vol. 2, no. 4, pp. 446-455, Dec. 1994.

[4] T. Chou and K. Roy, "Statistical estimation of sequential circuit activity," *Proc. Int. Conf. CAD*, pp. 34-37, 1995.

[5] T. Chou and K. Roy, "Accurate power estimation of CMOS sequential circuits," *IEEE Trans. on VLSI Systems*, vol. 4, no. 3, pp. 369-380, Sept. 1996.

[6] C. Tsui, J. Monteiro, M. Pedram, A. Despain, and B. Lin, "Power estimation methods for sequential logic circuits," *IEEE Trans. on VLSI Systems*, vol. 3, no. 3, pp. 404-416, Sept. 1995.

[7] F. N. Najm, S. Goel, and I. N. Hajj, "Power estimation in sequential circuits," *Proc. Design Automation Conf.*, pp. 635-640, 1995.

[8] S. Devadas, K. Keutzer, J. White, "Estimation of power dissipation in CMOS combinational circuits using boolean function manipulation," *IEEE Trans. CAD.*, pp. 373-383, March 1992.

[9] H. Kriplani, "Worst case voltage drops in power and ground busses of CMOS VLSI circuits," *Ph.D. Thesis, Univ. of Illinois*, Nov. 1993.

[10] H. Kriplani, F. Najm, P. Yang, and I. Hajj, "Resolving signal correlations for estimating maximum currents in CMOS combinational circuits," *Proc. Design Automation Conf.*, pp. 384-388, 1993.

[11] S. Manne, A. Pardo, R. I. Bahar, G. D. Hachtel, F. Somenzi, E. Macii, and M. Poncino, "Computing the maximum power cycles of a sequential circuit," *Proc. Design Automation Conf.*, pp. 23-28, 1995.

[12] C. Wang, K. Roy, and T. Chou, "Maximum power estimation for sequential circuits using a test generation based technique," *Proc. Custom Integrated Ciruits Conf.*, 1996.

[13] D. G. Saab, Y. G. Saab, and J. A. Abraham, "CRIS: A test cultivation program for sequential VLSI circuits," *Proc. Int. Conf. CAD*, pp. 216-219, 1992.

[14] E. M. Rudnick, J. H. Patel, G. S. Greenstein, and T. M. Niermann, "Sequential circuit test generation in a genetic algorithm framework," *Proc. Design Automation Conf.*, pp. 698-704, 1994.

[15] P. Prinetto, M. Rebaudengo, and M. Sonza Reorda, "An automatic test pattern generator for large sequential circuits based on genetic algorithms," *Proc. Int. Test Conf.*, pp. 240-249, 1994.

[16] M. S. Hsiao, E. M. Rudnick, and J. H. Patel, "Automatic test generation using genetically-engineered distinguishing sequences," *Proc. VLSI Test Symp.*, pp. 216-223, 1996.

[17] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning,* Reading, MA: Addison-Wesley, 1989.

[18] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design.* New York, NY: Computer Science Press, 1990.

[19] T. M. Niermann and J. H. Patel, "HITEC: A test generation package for sequential circuits," *Proc. Euro. Conf. Design Automation (EDAC)*, pp. 214-218, 1991.

[20] T. M. Niermann and J. H. Patel, "Method for automatically generating test vectors for digital integrated circuits," *U.S. Patent No.* 5,377,197, Dec. 1994.

[21] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," *Int. Symp. on Circuits & Systems*, pp. 1929-1934, 1989.