Peak Power Estimation of VLSI Circuits: New Peak Power Measures

Michael S. Hsiao, Elizabeth M. Rudnick, and James H. Patel

Abstract—New measures of peak power are proposed in the context of sequential circuits, and an efficient automatic procedure is presented to obtain very good lower bounds on these measures, as well as providing the actual input vectors that attain such bounds. Automatic generation of a functional vector loop for near-worst case power consumption is also attained. Experiments show that vector sequences generated give much more accurate estimates of peak power dissipation and are generated in significantly shorter execution times than estimates made from randomly generated sequences for four delay models.

Index Terms—Delay models, peak power, power consumption, power estimation, VLSI design.

I. INTRODUCTION

Peak power estimation enables circuit designers to optimize a circuit in order to avoid circuit failures due to peak-power-related hazards. In this paper, a peak power estimation tool, K2, is described that generates a specific vector sequence that produces maximum power dissipation in a circuit for both combinational and sequential circuits. Furthermore, the effects of different delay models on the estimation of peak power dissipation are discussed.

Power dissipation in CMOS circuits has two components: static, due to leakage current; and dynamic, due to switching activity. The static power is relatively low and is often neglected in power estimation. 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. Throughout this paper, when we refer to power, we will mean the capacitive component of dynamic power.

Estimating peak power involves maximization of a circuit’s switching function. The problem is further complicated by the fact that switching of a gate is heavily dependent on the gate delays, since multiple transitions can occur at internal nodes due to uneven delay paths in the circuit. Finally, the initial state of a sequential circuit is an important factor in determining the amount of switching activity and must be taken into account.

The problem of estimating the worst case power dissipation in CMOS combinational circuits has been addressed in [1], in which the worst case power is transformed into a weighted max-satisfiability problem. Peak current estimation for combinational circuits is addressed in [2], [3], and the goal is to find the time window during which a gate in the circuit could switch. Maximum power cycles are computed using symbolic transition counts in [4]; the state transition diagram (STG) is used to find the maximal average cycle in the graph. Peak power estimation is computed for sequential circuits using test generation based techniques in [5] and [6]. Attempts are made to create toggles in the circuit for gates with the greatest numbers of fanouts. Different delay models are used in [6], in which the circuit is expanded by including multiple copies of internal gates to account for

Manuscript received July 31, 1998; revised March 4, 1999 and August 25, 1999.

M. S. Hsiao is with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854-8058 USA (e-mail: mhsiao@ece.rutgers.edu).

E. M. Rudnick and J. H. Patel are with the Center for Reliable and High-Performance Computing, University of Illinois, Urbana, IL 61801 USA (e-mail: liz@crhc.uiuc.edu; patel@crhc.uiuc.edu).

Publisher Item Identifier S 1063-8210(00)01035-0.
the propagation delays of the gates. Finally, a genetic algorithm (GA) was used to estimate maximum power in [7], but results were reported only for single-cycle maximum power.

In our work, new measures of peak power are proposed in the context of sequential circuits, and an automatic procedure is developed to very quickly obtain tight lower bounds for these measures, as well as providing actual input vectors that attain such bounds. Automatic generation of a functional vector loop for near-worst case power consumption is also attained. Furthermore, the initial state of the circuit in sequential circuits is taken into account in this measure. Moreover, for the pair of vectors to be useful, the initial state needs to be reachable. In this work, we can obtain peak power with or without the restriction of reachability of the state \( S_1 \). 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.

Peak \( n \)-cycle power is illustrated in Fig. 1(b). We will search for an \((n + 2)\)-tuple \((S_1, V_1, \cdots, V_n, V_{n+1})\) 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. Single-cycle power is close to the instantaneous peak, 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 valid state as in single-cycle power.

Peak sustainable power is illustrated in Fig. 1(c). The state \( S_1 \) is repeated at the end of the sequence. 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

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

where \( \text{toggle}(g) \) is the number of switches (0–1 or vice versa) for gate \( g \) in a clock 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 rewritten as

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

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

Four different delay models are studied in this work: zero delay, unit delay, type-1 variable delay, and type-2 variable delay. Zero-delay assumes no delays for any circuit elements, unit-delay assigns equal delay to every circuit element, and type-1 variable delay assigns a delay for a gate that is proportional to the number of gate fanouts. This model is more accurate than the unit delay model; however, fanouts that feed bigger gates are not taken into account, and inaccuracies may result.

This is where the fourth model comes in. It is a variable delay model based on the fanouts as well as the sizes of successor gates. The gate delay data for various types and sizes of gates are obtained from a VLSI library. Type-2 variable delay takes the input capacitance, in addition to output capacitance of a gate, into consideration.

A. Genetic Algorithm Framework

The GA framework used in the implementation of the algorithm K2 is similar to the simple GA described in [8] and [9]. Peak $n$-cycle power estimation requires a search for the $(n + 2)$-tuple $(S_1, V_1, \ldots, S_n, V_{n+1})$ 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 vector sequences in order to maintain diversity. However, linear scaling with the input space is not feasible. The population size is set equal to $32 \times \sqrt{\text{sequence length}}$ when the number of primary inputs is less than 16 and $128 \times \sqrt{\text{sequence length}}$ when the number of primary inputs is greater than or equal to 16. The sequence length parameter of the equation is set to 2 for combinational circuits because only two vectors are needed to set up the state of the circuit and measure the activity induced by the second vector. Since the majority of time spent by the GA is in the fitness evaluation, parallelism among the individuals can be exploited. Parallel-pattern simulation [10] is used to speed up the process; thus, 32 individuals from the population are simulated simultaneously, with values bit-packed into 32-bit words.

III. PEAK SINGLE CYCLE AND $N$-CYCLE POWER ESTIMATION

We estimate the power dissipation in CMOS circuits by measuring the amount of switching activity; static power dissipation is neglected. The vector sequence that produces peak single-cycle power may not simultaneously generate the peak $n$-cycle power, or vice versa. 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, 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. The peak levels off as the sequence length approaches infinity or earlier when a loop is found that is capable of maintaining the given power.

In order to take the initial state into consideration during power estimation, the state portions of the individuals in the GA are seeded using a previously computed set of reachable states of the circuit, $S_{reach}$ (this set may not be complete). We compute $S_{reach}$ in a preprocessing step using a GA in which the fitness function is set to visit as many new states as possible. The GA initially contains random strings, and the evolutionary process aims to maximize the number of new states visited by the individuals in the population. Because our goal is to maximize the set of states reached, the fitness of an individual is simply the number of new states visited. This process terminates when a specified number of states has been visited (10,000 for large circuits).

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 circuit test generation state justification procedure. State justification, however, is a very complex problem involving a large number of backtracks [11]. In order to reduce the execution time, an alternative to state justification is taken. A set of states, $S_{init}$, is formed by selecting states from $S_{reach}$, that are similar to the required target state. Thus, every state in $S_{init}$ is also a reachable state. A state $S_j$ is similar to another state $S_i$ if the Hamming distance between their encodings is short. The best vector-sequence generated by the GA is simulated from every state in $S_{init}$, and the maximum power obtained from the set of states is taken as the peak power. For most circuits, the difference in peak powers for the desired starting state and the selected similar starting state is very small. Again, parallelism is exploited among the different starting states during simulation.

IV. PEAK SUSTAINABLE POWER ESTIMATION

The lower-bound for peak power dissipation reaches a steady state when the sequence length goes to infinity. The term peak sustainable power denotes the steady-state value. Consider the case where state $A$, together with vectors $V_i$ and $V_j$, is a three-tuple, i.e., $(A, V_i, V_j)$, the peak single-cycle power dissipation for the chip; in addition, state $A$ is repeated by applying vectors $V_j$ and $V_j$. If a vector sequence $T_{init}$ initializes the circuit to state $A$, and the $(V_i, V_j)$ input-pair is repeated and concatenated to $T_{init}$, the peak power can be sustained for every time frame after state $A$ has been reached, and this sequence becomes a peak-power-sustaining sequence. Unfortunately, it is non-trivial to find such loops, even if we were working with a state transition graph.

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. Another approach is to derive a peak $n$-cycle power sequence starting from an easily reachable state $S_{neg}$, then close the loop with only a few additional transitions. We take this approach to an extreme, starting with the entirely unknown state. Since this state is a superset of any state, it can be reached in just one transition from any state. The sequence starting from the all-unknown state always forms a loop because the final state of the sequence is covered by the initial state. If the final state is a fully specified state, the sequence is also a synchronizing sequence. In fact, any synchronizing sequence is a loop. 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 show that this approach still yields peaks higher than extensive random search.

Generating a complete set of synchronizing sequences for a sequential circuit is not an easy task. Thus, techniques have been developed to generate a single or a small set of synchronizing sequences. They include the synchronizing tree method [12], binary decision diagrams (BDDs) method [13], and structural decomposition approach [14]. In our work, a GA is used to search for synchronizing sequences for a given circuit. Logic simulation is used for fitness evaluation in the GA, and the GA can stop as soon as a sequence of a given length is obtained that brings the circuit from a completely unspecified initial state to a completely specified final state. In trying to generate a synchronizing sequence of a given length $f$, the GA quits as soon as an acceptable sequence is found. If no synchronizing sequence can be found with length $f$, the GA doubles its sequence length to $2f$ and attempts to derive a sequence within the new length. This process terminates when a maximum length is reached.
The power estimation algorithms presented were implemented in a tool called K2 using the C++ language: ISCAS85 combinational benchmark circuits, ISCAS89 sequential benchmark circuits, and several synthesized circuits were used to evaluate the speed and accuracy of K2 on an HP 9000 1200 with 256 MB of RAM. K2 derives synchronizing sequences of equal lengths to those in [14] in all circuits except $s_{5378}$, for which a shorter synchronizing sequence was derived. For most circuits, K2 took more execution time due to numerous logic simulations for the individuals in the population; however, these execution times are all very short (all less than 3 s, except for $s_{1423}, s_{5378},$ and $s_{53932}$, where up to 40 s were needed). This experiment shows that genetic algorithms are able to find a solution even when the solution is nontrivial, such as in the case of $s_{5378}$.

All power estimates from K2 were compared against the estimates obtained from randomly generated vector sequences for all four delay models. However, only results for unit-delay are reported, since the powers obtained using type-1 and type-2 variable delay models are similar [15]. In addition, the peak single-cycle power estimates for sequential circuits are compared against the results obtained in [6] using a unit-delay model. Note that results are reported for the zero-delay model only in [5]; thus, they are not listed. We express all power measures in peak switching frequency per node (PSF), which is the average frequency of peak switching activity per node (ratio of the total number of 0-1 and 1-0 transitions to the total number of capacitive nodes) in the circuit. Because multiple switches on internal nodes can occur within one clock cycle for nonzero-delay models, PSFs greater than 1.0 are possible.

To verify that the estimates made by K2 are indeed good estimates of peak power, extensive simulations were performed using the unit-delay model. Millions of random state-vector tuples were simulated for seven small sequential circuits: $s_{298}, s_{382}, s_{400}, s_{444}, s_{526}, s_{641},$ and $s_{713}$. For comparison, if an exhaustive search was performed for $s_{400}$, which has 21 flip-flops and three primary inputs, the total number of simulations for the three-tuple ($s_{1}, V_{1}, V_{2}$) would be $2^{(21+3+3)} = 2^{27} \approx 134$ million. For most circuits, peak single-cycle power estimated by K2 (which required less than one minute per circuit) is equal to or slightly greater than the estimate made from 100 million random vector-pairs, which took many hours of simulation time. For the last two circuits, $s_{641}$ and $s_{713},$ the results from simulation of 100 million random sequences (over 22 h of execution) still lag 4.4% behind K2 estimates. Pseudo-exhaustive simulations for peak $N$-cycle and sustainable powers were not conducted, since they would require billions of state-vector tuples for even the small circuits. From this experiment, we observe that if random-based methods were to achieve similar tightness of peak power bounds, several orders higher execution times would be needed.

Proceeding to the experiments on peak power, Table I compares the K2 results against the randomly generated sequences as well as those reported in [6] for the unit-delay model for the ISCAS85 combinational circuits. In our experiments, we restrict the random search to use the same number of simulations used by our tool K2 for each circuit. Typical numbers of simulations exceed 64,000. It is difficult to compare our results with those in [7] since the underlying delay model used in [7] was not described. For each circuit, results for the best of randomly-generated vector-pairs, an ATPG-based approach [6], and K2 are shown. The estimates made by K2 are the highest for every circuit. The average improvements made by K2 over the random simulations was 27.4%.

The K2 estimates for sequential circuits under the unit-delay model are shown in Table II. For each circuit, the peak power obtained by the random approach and K2 are given in terms of PSF. K2 execution times are also reported for each circuit in seconds. For the smaller circuits, the number of simulations is about 64,000. Execution times for the random approach are similar since the number of simulations is identical; however, they may sometimes be slightly lower due to fewer events occurring. For peak sustainable power, the random search merely searches for a sequence that has at least one state repeated in the sequence and produces the most power among all random sequences. The average improvements achieved by K2 are shown at the bottom of the table. The estimates made by K2 surpass the best random estimates for all circuits in all three peak power measures. Up to 32.5% improvement is obtained for peak ten-cycle powers. The execution times are directly proportional to the number of events generated in the circuit during the course of estimation. For this reason, peak ten-cycle power estimates do not take ten times as much computation as peak single-cycle power, since the amount of activity across the ten cycles is not ten times that of the peak single cycle.

In many circuits, significant gaps between the peak single-cycle and ten-cycle power estimates exist, indicating that the peak single-cycle power dissipations are difficult to sustain for these circuits. However, some circuits have peak sustainable power estimates very near their corresponding peak ten-cycle power lower-bounds; this suggests that the power estimated for the peak ten-cycle power can be sustained.

The peak single-cycle and ten-cycle power estimates discussed so far are valid only if the initial states for these sequences are reachable. Reachability analysis was performed for the initial state, and the respective power dissipations were compared for estimates computed before and after the reachability analysis. The time required to perform reachability analysis using the similar states approach described in Section III was negligible. Reachability analysis is not needed for peak sustainable power because sustainable power is estimated using synchronizing sequences that take the circuit from the all-unknown state to a known state. 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 likely to be unreachable for circuits with many flip-flops. For these circuits, a lower peak power measure results after the reachability analysis. However, there are cases where the power dissipations increase slightly when sequences are applied starting from reachable states that had not been considered previously in the huge search space. In all cases, the estimates computed by K2 after reachability analysis are still higher than the best estimates from randomly-generated vector-pairs after reachability. Average drops of 17.6% and 20.1% in peak power are observed after the reachability analysis for the random and K2 peak single-cycle powers, respectively. The drops are insignificant for many circuits, while in a few circuits, such as $s_{641},$ $s_{713},$ $s_{53932},$ and mut16, the differences are greater. For the peak

<table>
<thead>
<tr>
<th>Ckt</th>
<th>Rand</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>1.985</td>
<td>1.985</td>
</tr>
<tr>
<td>c499</td>
<td>1.268</td>
<td>0.950</td>
</tr>
<tr>
<td>c880</td>
<td>1.367</td>
<td>0.995</td>
</tr>
<tr>
<td>c1355</td>
<td>1.681</td>
<td>1.268</td>
</tr>
<tr>
<td>c1908</td>
<td>2.641</td>
<td>1.961</td>
</tr>
<tr>
<td>c2670</td>
<td>1.690</td>
<td>1.676</td>
</tr>
<tr>
<td>c3540</td>
<td>2.351</td>
<td>1.646</td>
</tr>
<tr>
<td>c5315</td>
<td>1.859</td>
<td>-</td>
</tr>
<tr>
<td>c6288</td>
<td>19.301</td>
<td>-</td>
</tr>
<tr>
<td>c7552</td>
<td>2.052</td>
<td>-</td>
</tr>
</tbody>
</table>

| Impr | 27.4% |

TABLE I

PEAK UNIT-DELAY POWER FOR ISCAS85 CIRCUITS
ten-cycle powers, a drop in peak power is still observed; however, this time, the drop is only 7.2%. Again, the drops in power are mainly due to a few circuits where the state space is enormous.

VI. CONCLUSIONS

A GA-based power estimation framework, $K_2$, was developed. New measures of peak power in the context of sequential circuits are proposed. Estimates for peak single-cycle, ten-cycle, and sustainable power dissipation are computed, and vector sequences that attain such powers are also derived. The role of the initial state in sequential circuits has been taken into account. $K_2$ was shown to be very effective in deriving synchronizing sequences, which are then used to compute peak sustainable powers. $K_2$ generates vector sequences that obtain much tighter bounds when compared to the estimates made from randomly-generated sequences and previous approaches [5], [6].

The average improvements in the estimates for $K_2$ to the best of the previous two approaches are 13.5% for combinational circuits and 13.4% for sequential circuits. In addition, the execution times of $K_2$ are orders of magnitude lower than those for random-based estimates, if they are to achieve similar tightness of lower bounds, especially for larger circuits.

REFERENCES


