

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220730066>

A novel ACO-based pattern generation for peak power estimation in VLSI circuits

Conference Paper · March 2009

DOI: 10.1109/ISQED.2009.4810314 · Source: DBLP

CITATIONS

7

READS

421

4 authors, including:



Chun-Yao Wang

National Tsing Hua University

101 PUBLICATIONS 665 CITATIONS

SEE PROFILE



Yung-Chih Chen

147 PUBLICATIONS 1,922 CITATIONS

SEE PROFILE

A Novel ACO-based Pattern Generation for Peak Power Estimation in VLSI Circuits

Yi-Ling Liu, Chun-Yao Wang, Yung-Chih Chen and Ya-Hsin Chang
Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
g9562548@oz.nthu.edu.tw, {wc Yao, ycchen, br9662520}@cs.nthu.edu.tw

Abstract

Estimation of maximal power consumption is an essential task in VLSI circuit realizations since power value significantly affects the reliability of the circuits. The key issue of this task is which pattern pair would cause this peak power value. An exhaustive search from all possible combinations is time-consuming and impractical for VLSI circuits with hundreds of inputs. In this paper, a new pattern generation approach with *Ant Colony Optimization*, which imitates the behavior of ants looking for foods, is proposed for peak power estimation. The approach returns patterns which are highly suspicious to consuming the peak power. The gate delay issue is considered in this work. Furthermore, the valid state issue in sequential circuits is considered as well. Since the real delay value is technology-dependent, these generated patterns of our approach and other approaches are then applied into a commercial power calculation tool, PrimePower, to demonstrate the effectiveness of the approach under the TSMC 0.18 μm and TSMC 0.13 μm library. The experimental results show that an average of 76% tighter lower bound for the ISCAS'85 combinational benchmarks and 52% for the ISCAS'89 sequential circuits are obtained as compared to random patterns under the TSMC 0.18 μm library. For TSMC 0.13 μm library, our patterns obtain 90% and 56% improvements over the random patterns for ISCAS'85 and ISCAS'89, respectively. As compared with the previous work, our results are also competitive.

Keywords

Ant Colony Optimization (ACO), peak power estimation

1. Introduction

With the continuing increase of chip density and the shrink of feature size in VLSI circuits, the excessive power dissipation has become a concerned issue. Excessive power dissipation can cause not only overheating, which leads to a reduction of the life-time of a chip, but also the failure rate of the components, which doubles for every 10°C increase in operating temperature [15]. In addition, the use of portable devices, request for low-power dissipation, has been rapidly increasing.

In CMOS logic circuits, the power dissipation is closely relative to the factors of clock frequency, gate delays, gate output capacitances, circuit architectures, process parameters, and input patterns applied. Once the process parameters and the circuit architecture are determined, the power dissipation is dominated by the switching activity (toggle counts), which are related to gate delays, gate output capacitances, and input patterns of the circuit [7]. In particularly, the gate delays strongly influence the peak power dissipation. As pointed in [7], the peak power consumption with the zero-delay model

is extremely different from that with the accurate delay model since the behavior of glitches and hazards are not taken into account under the zero-delay simulation.

Another category related to the peak power estimation is the Maximum Instantaneous Current (MIC) estimation since power is a function of current. Many efforts have been invested in this problem [2] [6] [10]. The MIC is considered as the upper bound of actual current. In this paper, however, the focus is on the lower bound of peak power estimation, since the upper bound value could be pessimistic.

One way to find the vector pair with the peak power dissipation is to exhaustively search from all possible input pattern combinations. For a circuit with n primary inputs, the overall patterns required are up to $(2^n)^2 = 4^n$. Thus, this approach is only feasible for small circuits.

Although many efforts have been invested in the estimation of power dissipation in [1] [5] [12] [13] [17], they focus on the average power. The average power is typically used as a reference for designing the power networks in the ICs, and this power value usually can be obtained from the calculation with signal switching probability. For peak power estimation, however, it needs to find out a specific pattern pair in the large input space to get a tighter lower bound. This problem has been invested in [3], and is transformed into a complex NP-complete max-satisfiability problem, where a logic description is converted into a multiple-output Boolean function. It then deals with the problem with either a disjoint cover enumeration technique or a branch-and-bound algorithm to obtain the vector which is suspicious to peak power. Once this input vector or a vector sequence has been determined, circuit-level simulation is performed to accurately determine the associated power dissipation. Nevertheless, this algorithm could only deal with small circuits because of the high computation complexity.

The method of computing the maximum power cycles with the symbolic transition counts is addressed in [11]. The State Transition Graph (STG) is used to compute the maximum average cycles in the graph, and the power dissipation between two adjacent states in the STG is computed to simultaneously derive the maximal power consumption. The approach, however, could only deal with small circuits due to the memory problem, e.g., the test case s208 has a dual graph with 34816 vertices and more than 71 million edges. Also, two matrices have to be stored in computation, and the memory requirements exceed 1 GBytes.

Another test generation-based technique is proposed in [19] to search the maximum transition nodes in the circuit. It looks for the nodes with a large number of fanouts, and then backward justifies to the primary inputs to get the patterns. However, this method is inaccurate with the zero-delay model, which neglects the power contributed by glitches and hazards.

Another category about the peak power estimation is proposed in [8] [9]. These work both exploit the Genetic Algorithm (GA) to produce different patterns under several delay models. Then a small set of patterns associated with the peak power are derived. The Peak Switching Frequency (PSF) is used to represent the peak power consumption in the circuits. However, the correlation between the PSF value and the actual peak power is not explored in the paper, and the process technology is not taken into account, either. In addition, the valid state issue in sequential circuits is not considered. Thus, the reported PSF values are overestimated in the sequential circuits if the associated state is unreachable under the normal operations.

In this paper, the Ant Colony Optimization (ACO) [4] technique is applied for deriving the patterns with high PSF values in VLSI circuits. This method can deal with large combinational and sequential circuits with considering general delay models [14]. Since the delay model used is still not accurate, the generated patterns are then applied into a commercial tool, PrimePower [16], for the actual power calculation. The results returned by PrimePower can be used to demonstrate the effectiveness of our approach when comparing with other approach. The experimental results show that the maximal power consumption of the produced patterns is much higher than that of random patterns under the same amount of CPU time.

The remainder of this paper is organized as follows. Section II briefly introduces the problem formulation and the ACO algorithm. Section III details our ACO-based approach to this problem. Section IV describes the overall flow. Section V shows the experimental results of our approach. Section VI concludes the paper.

2. Preliminaries

2.1. Problem Formulation

The dissipated power in sequential circuits is governed by both initial state S_I and input vectors V_I , V_2 as shown in Fig. 1 [9]. The initial state S_I and input vector V_I are used to initialize the internal nodes, and the state will change to S_2 . After the application of input vector V_2 under the new state S_2 , some transitions that contribute to the power consumption occur in the internal nodes. A path that attempts to search the peak power dissipation from a valid initial state is illustrated in Fig. 2. The $2n$ -cycle path aims for finding one cycle that contributes the maximal power consumption in the whole path.

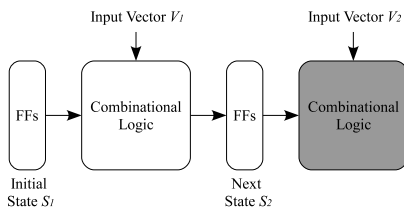


Fig. 1. The single cycle power model for sequential circuits.

For the computation of power consumption, let's consider a circuit in Fig. 3. During the time interval of $(0, +T]$, four

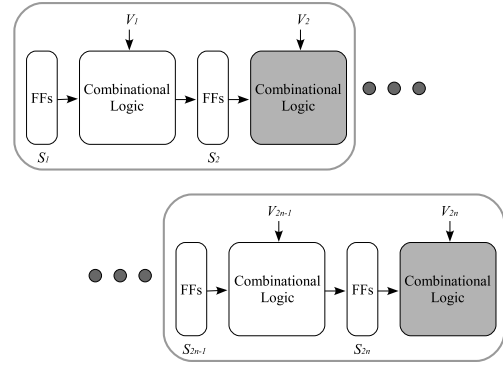


Fig. 2. Estimation of peak power consumption from a valid initial state S_I .

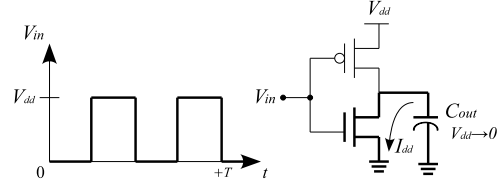


Fig. 3. An inverter circuit for finding the power dissipation.

transitions (rising or falling) occur, where T is the cycle time, V_{dd} is the supply voltage, and C_{out} is the gate output capacitance, the consumed power is derived as Equation (1)

$$P = \frac{1}{2} \cdot I_{dd} V_{dd} \cdot 4 = \frac{1}{2} \cdot \frac{Q}{T} V_{dd} \cdot 4 = \frac{1}{2} \cdot \frac{C_{out} V_{dd}^2}{T} \cdot 4 \quad (1)$$

where Q is the amount of charges transferred. Since a complete cycle, rising and falling, will consume a unit of power (charging and discharging), the constant $\frac{1}{2}$ is introduced in Equation (1) to represent this fact. For a circuit with m internal nodes (gate outputs), if the charging and discharging currents are only considered, the average power dissipated in the time interval $(0, +T]$ for all internal nodes can be computed as Equation (2) [18]

$$P = \frac{V_{dd}^2}{2T} \sum_{i=1}^m C_i \times D_i \quad (2)$$

where C_i is the gate output capacitance at node i , and D_i is the number of rising and falling transitions at node i in the time interval $(0, +T]$.

According to Equation (2), the power consumption in a circuit is proportional to the amounts of $C_i \times D_i$ of all nodes. Thus, a Transition Density (TD) which is a function of $\sum_{i=1}^m C_i D_i$ is defined in Equation (3). It is an important parameter of our ACO-based peak power estimation algorithm. The total number of capacitive nodes in Equation (3) equals the total number of gate inputs.

$$TD = \frac{\sum_{i=1}^m C_i D_i}{\text{total number of capacitive nodes}} \quad (3)$$

In this paper, we assume that the output capacitance of a node C_i is equal to its fanout count. Nevertheless, the proposed technique can deal with the circuits having different output capacitance definitions.

2.2. Ant Colony Optimization

While ants walk between the nest and the food sources, they deposit chemical material named *pheromones* on the path forming a pheromone trail. Depending on the distance and the quality of food source, ants mark the path by adjusting the amount of pheromone that affects the probabilities of paths been walked. Ants tend to choose the paths marked with stronger pheromone concentrations. This mechanism of searching for food is known as Ant Colony Optimization (ACO) algorithm [4] and has been applied to solve many combinatorial optimization problems, e.g., the traveling salesman problem (TSP), a classic NP-hard problem. In general, ACO algorithm consists of three typical steps:

- 1). *Initial solution construction*: This step searches a feasible solution that is used to derive the final solution of the problem. It influences the quality of results of ACO algorithm.
- 2). *Pheromone update*: The pheromone values vary based on the quality of the trails found. The amount of pheromones on paths determine the probabilities of paths ants select. The more pheromones accumulated on a path, the higher probability the path chosen. As time passes, the pheromones also decrease. This mechanism is named *pheromone evaporation*, and it is used to alleviate the impacts of bad decisions been made. It also prevents the algorithm from rapid convergence and being stuck at suboptimal solutions.
- 3). *Local search*: The ACO algorithm also accompanies a local search to adjust the pheromone deposition to guide the search process.

2.3. Delay Model

Different cell delay models in the circuits lead to the disparity of power consumptions compared to the zero-delay model [7]. The general delay model [14] is used in this paper. However, since the actual cell delays are technology dependent and differ from a library to another library, the used delay model may not match the actual delay values. Thus, we augment the returned pattern set, e.g. 100 pattern pairs, such that the actual peak power pattern pair is included with higher probability. Finally, a power calculation tool, PrimePower, is used to calculate the actual power values of these pattern pairs to demonstrate the effectiveness of the pattern set.

3. ACO-based Peak Power Estimation

This section presents our approach to the peak power estimation. The peak power estimation for combinational circuits is described in Section III.A to III.C. Section III.D is for the sequential circuits.

3.1. Initial Solution Construction

Initial solutions affect the final results. To construct a better initial solution that has much more TD, 10,000 random pairs of patterns (V_I, V_2) are applied to the circuits. After the simulation by our program, the TD of each pattern pair can be calculated. This value indicates how this pair of patterns contributes to the power consumption.

For example in Fig. 4, assume the AND gates A and C are with 2-unit delays, and the OR gate B and D are with 3-unit delays. These delay values are also shown on the gates. When the input vector pair, (1110, 0100), is simulated, some

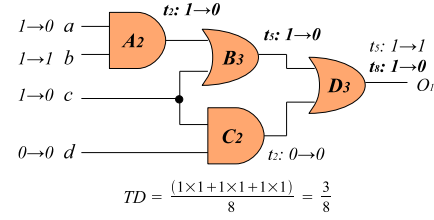


Fig. 4. The simulation process of the vector pair (1110, 0100).

transitions occur in these gates, e.g., the gate A switches from 1 to 0 at t_2 , but the gate C is intact. At t_5 , the gate B also switches. Subsequently, the TD of this vector pair is derived as Fig. 4 shows. The numerator of TD is the summation of the transition numbers multiplying the number of gate outputs of each node. The denominator of TD is the number of capacitive nodes which is the total number of gate inputs in this work. The TD value is proportional to the power consumption.

3.2. Pheromone Update

The pheromone is updated based on the quality of solution found. For each primary input i , each vector pair would cause four transition conditions $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$, and $1 \rightarrow 1$. Four parameters $\tau_{0 \rightarrow 0}^i$, $\tau_{0 \rightarrow 1}^i$, $\tau_{1 \rightarrow 0}^i$, and $\tau_{1 \rightarrow 1}^i$, which represent the pheromones with respect to these four conditions of each primary input i are then used to accumulate the TD of each input vector pair on the primary input i . The pheromone values also decay with a rate k under the mechanism of pheromone evaporation. The pheromone update behavior is heuristically expressed as Equation (4)

$$\tau_{0 \rightarrow 0}^i(t+1) = \Delta \tau_{0 \rightarrow 0}^i(t) + (1-k) \cdot \tau_{0 \rightarrow 0}^i(t) \quad (4)$$

where $k \in [0, 1]$ is the pheromone decay rate, and $\Delta \tau_{0 \rightarrow 0}^i(t)$ is the amount of pheromones added to the accumulated pheromone $\tau_{0 \rightarrow 0}^i$ at time t . The other three parameters are also expressed as Equation (4) except the subscripts of transition conditions.

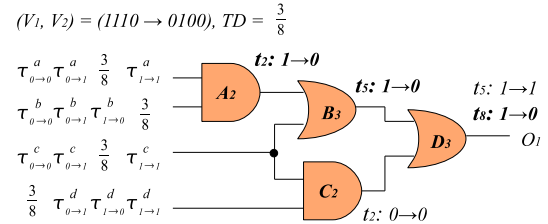


Fig. 5. The pheromone update process for the vector pair (1110, 0100).

Take Fig. 5 as an example that shows the same circuit as Fig. 4. Four parameters, $\tau_{0 \rightarrow 0}^a$, $\tau_{0 \rightarrow 1}^a$, $\tau_{1 \rightarrow 0}^a$, and $\tau_{1 \rightarrow 1}^a$, represent the pheromones of four transition conditions of one vector pair on the primary input a . These parameters are initialized to zero. After the input vector pair $(V_I, V_2) = (1110, 0100)$ is applied, the TD of this vector pair is derived as Fig. 4. Then, this value $\frac{3}{8}$ is added to τ^i with respect to the transition conditions associated with the vector pair. For instance, the transition condition of the primary input b between (V_I, V_2) is $1 \rightarrow 1$. Thus, the parameter $\tau_{1 \rightarrow 1}^b$ is increased by the TD $\frac{3}{8}$. These pheromone parameters will

be continually updated for each input pair. After simulating 10,000 random vector pairs, two kinds of data are obtained for the next step. The first one is the top 100 vector pairs that have higher TDs. The second one is the accumulated pheromones on all primary inputs.

3.3. Local Search

A local search procedure is then applied to search better solutions. After acquiring the results of 10,000 initial vector pairs, the top 100 vector pairs with higher TDs are chosen for local search. In addition, for each primary input, the first two highest values of these four pheromone parameters are randomly selected for generating another 100 vector pairs that will be crossovered with those top 100 vector pairs. The purpose of picking up the first two highest parameters is to improve the varieties of vector pairs.

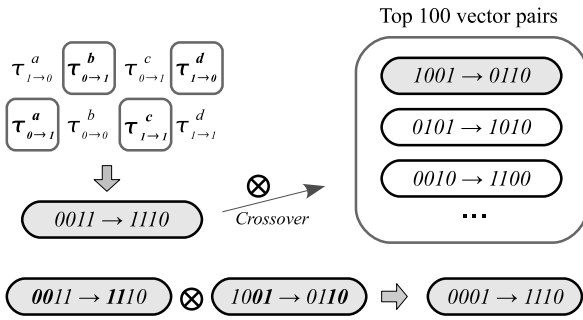


Fig. 6. The local search process in our ACO algorithm.

As illustrated in Fig. 6, assume the upper row of the left side lists the pheromones of the highest value, and the lower one is the second highest parameters. One vector pair will be generated by randomly choosing one parameter, either the highest or the second highest. In this figure, the vector pair ($0011 \rightarrow 1110$) is generated that is derived from a, c of the second row, and b, d of the first row (squared). This new generated vector pair is pairwise crossovered with a vector pair from the right side of Fig. 6 sequentially, which is the set of top 100 vector pairs. The crossover operation concatenates the first half of the left vector pair and the second half of the right vector pair to form another vector pair as illustrated at the bottom of Fig. 6. After getting new crossovered 100 vector pairs, we simulate them again and keep updating the pheromones. The local search will continue until the solution is intact, or it terminates after 32 iterations.

3.4. Sequential Circuits

The difference between combinational circuits and sequential circuits is the flip-flop issue. The states of sequential circuits can be assigned to arbitrary values only under the full-scan mode of operation, which is used for testing. If the initial state could be assigned to an arbitrary value during the peak power estimation, the power value will be overestimated since unreachable states are illegally allowed. Thus, this work starts from a given reset state, which is reachable, and aims to find one path that contains a cycle with the peak power consumption. A sequential circuit can be considered as a series of combinational circuits with different states from the

viewpoint of timeframe expansion.

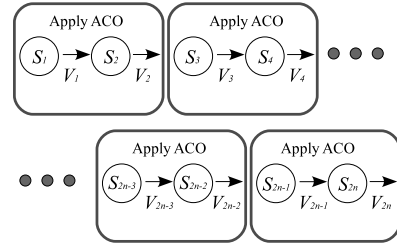


Fig. 7. Apply ACO algorithm for sequential circuits.

The legal peak power occurs in a cycle only under a reachable state. Thus, the power consumption in one cycle can be calculated by deriving the tuple (S_i, V_i, V_{i+1}) . As illustrated in Fig. 7, since S_1 is the reset state, the power consumption at this state can be attained by observing the transitions of nodes with applying a (V_1, V_2) vector pair. The vector pair (V_1, V_2) that consumes the maximal power is also derived by the proposed ACO algorithm under state S_1 . The vector pairs (V_3, V_4) to (V_{2n-1}, V_{2n}) in the following cycles are derived in the same manner. Thus, we can get a path from the reset state to an internal state such that each state transition accompanies a local peak power. These vector pairs associated with the states are considered as the candidates of peak power. Then we choose a larger one as the peak power of the circuit after applying these vector pairs to the PrimePower tool. Note that all states in the path are valid since they are reached from the reset state.

4. Actual Peak Power Calculation and Overall Algorithm

To acquire the actual maximal power consumption with considering the process technology, a power calculation tool, PrimePower [16], is used with the generated vector pairs. Section IV.A briefly introduces the tool. Section IV.B describes the overall algorithm.

4.1. Power Calculation

PrimePower is a simulation-based gate-level power analysis tool that accurately analyzes power dissipation of cell-based designs. It builds a detailed power profile of the design based on the circuit connectivity, the switching activities, the net capacitances and the cell-level power behavior data in the .db library.

The power calculation procedure is comprised of two phases, and they are switching activity generation and power analysis. The overall flow with Value Change Dump (VCD) files is shown in Fig. 8. In the phase 1, the vector pairs generated from the proposed ACO algorithm are considered as the testbenches that will trigger the transitions of signals in the circuits. Combined with the synthesized design and the cell library, the testbenches are then simulated and the VCD files are obtained. In the phase 2 with the VCD files, PrimePower analyzes and reports the power consumption of the circuit associated with the testbenches and the library over the simulation period.

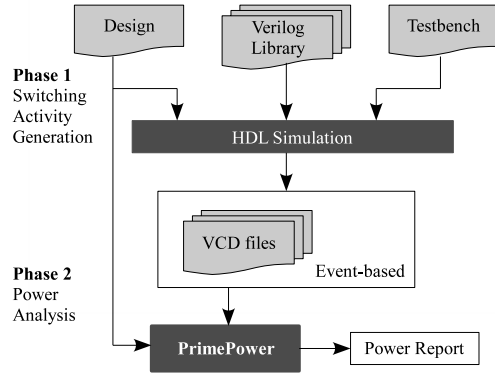


Fig. 8. Power calculation with the PrimePower.

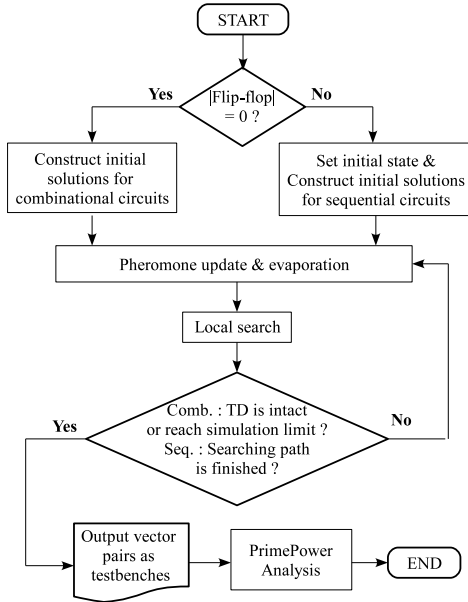


Fig. 9. The overall ACO algorithm.

4.2. Overall Algorithm

The overall ACO algorithm for estimating the maximal power consumption is shown in Fig. 9. It both deals with combinational and sequential circuits.

For the combinational circuits, random vector pairs are applied for initial solution construction, and the phormone values are updated based on the initial solutions. The local search procedure generates patterns by referring to the phormone parameters of each input, and then crossovers with the vector pairs having higher TD values. If the TD values are intact or the number of simulated vector pairs reaches the limit, the final top 100 vector pairs are returned as the testbenches.

For the sequential circuits, initial vector pairs are randomly generated in the reset state. The phormone parameters are updated based on the initial solution. Then, the local search procedure is applied as mentioned. Finally, the vector pair that contributes the maximal TD value is chosen for this cycle. This process is repeated for the succeeding cycles until the search path reaches a predefined cycle limit. These selected vector pairs are considered as the testbenches for the peak

power estimation with the PrimePower tool.

5. Experimental Results

In this section, the experimental results of the ACO algorithm for peak power estimation are presented on a set of ISCAS'85 combinational benchmarks and ISCAS'89 sequential benchmarks. The program is implemented in C++ language. The experimental platform is a Sun Blade 2500 workstation with 4GBytes memory. All of these benchmarks are then synthesized with the TSMC 0.18 μ m and 0.13 μ m libraries, and the vector pairs produced by a random approach and our ACO approach are applied to PrimePower for actual power calculation. The random approach simulates the same amount of CPU time as ours, and is for demonstrating the effectiveness of our approach.

For the combinational circuits, 10,000 random patterns are simulated for initial solution construction. Then, the local search will repeat 32 iterations, and each iteration contains 100 vector pairs with the phormone evaporation rate of 0.1. To avoid being trapped in the local optimum, another 10,000 random pairs will be applied after the local search procedures and repeat the local search until reaching the terminating condition. At last, 100 vector pairs with the highest TD values are returned, and applied to the PrimePower. According to the results returned by the PrimePower, the highest power value is recorded.

Table I shows the experimental results of our approach and the random approach for ISCAS'85 combinational benchmarks. Column 1 lists the benchmarks. Column 2 lists the number of primary inputs in each benchmark, PI . Columns 3 and 4 list the number of gates, NG , and the number of capacitance nodes, NC , in each benchmark, respectively. Columns 5 and 6 list the peak power values measured in mW with the TSMC 0.18 μ m library obtained by random patterns and ACO patterns under the same amount of CPU time as shown in Column 11, respectively. Column 7 lists the ratio of the results produced by the ACO and the random approaches. Columns 8 and 9 also list the peak power values with the TSMC 0.13 μ m library. Column 10 lists the ratio of the results produced by the ACO and the random approaches. Column 12 lists the number of random patterns generated within the CPU time shown in Column 11. For example, c3540 has 50 PIs, 1742 gates, and 2961 capacitance nodes. The peak power under the TSMC 0.18 μ m library is 65.1 mW for the random approach and is 237.1 mW for the ACO approach. The ratio of ACO and random approach is 3.65. The number of random patterns are 54202546. According to Table I, on average, the ACO approach gives 76% tighter lower bound of maximal power dissipation than that obtained by random patterns under the TSMC 0.18 μ m library. The ACO results also reach 90% tighter lower bound than that of the random approach under the TSMC 0.13 μ m library.

Table II shows the experimental results on ISCAS'89 sequential benchmarks. Column 5 lists the number of flip-flops in each circuit. Other columns are the same with Table I. The iteration of sequential path search is set to 20 in our experiments, which equals to 40 cycles. According to Table II, on average, the ACO approach gives 52% and 56% tighter lower bound than that obtained by random patterns under the

TABLE I
PEAK POWER ESTIMATION FOR ISCAS'85 COMBINATIONAL CIRCUITS.

Circuit	PI	NG	NC	RAN ₁₈	ACO ₁₈	ACO ₁₈ RAN ₁₈	RAN ₁₃	ACO ₁₃	ACO ₁₃ RAN ₁₃	Time (Sec.)	# of random patterns
c432	36	204	343	10.9	22.3	2.05	3.34	6.82	2.04	22.2	10972076
c499	41	276	440	10.48	17.9	1.71	3.1	5.29	1.71	55.4	24987156
c880	60	470	755	37.2	71.7	1.93	10.59	21.47	2.03	47.2	15946816
c1355	41	620	1096	27.9	29.45	1.06	8.53	18.23	2.14	34.8	15692067
c1908	33	939	1523	15.27	18.47	1.21	3.39	4.6	1.36	60.4	31934691
c2670	234	1567	2216	114.8	241	2.10	33.06	69.53	2.10	241.5	24816697
c3540	50	1742	2961	65.1	237.1	3.65	20.83	69.82	3.35	141.1	54202546
c5315	178	2609	4509	131.4	172.3	1.31	37.8	49.97	1.32	429.3	56110842
c6288	32	2481	4832	204.6	233.4	1.14	44.19	67.58	1.53	2814.3	506365098
c7552	207	3828	6252	240.8	338.8	1.41	70.41	102.34	1.45	1787.4	205997230
Average	-	-	-	-	-	1.76	-	-	1.90	-	-

TABLE II
PEAK POWER ESTIMATION FOR ISCAS'89 SEQUENTIAL CIRCUITS.

Circuit	PI	NG	NC	FFs	RAN ₁₈	ACO ₁₈	ACO ₁₈ RAN ₁₈	RAN ₁₃	ACO ₁₃	ACO ₁₃ RAN ₁₃	Time (Sec.)	# of random patterns
s344	9	101	295	15	9.46	11.95	1.26	2.58	3.52	1.36	58.2	79906670
s386	7	118	359	6	2.97	2.97	1.0	1.19	1.19	1.0	33.6	51542220
s1196	14	388	1045	18	23.4	39.72	1.7	7.18	12.29	1.71	124.2	134406574
s1423	17	490	1300	74	13.29	18.47	1.39	4.26	5.15	1.21	133.2	128062093
s1488	8	550	1410	6	17.18	25.73	1.5	5.05	7.95	1.57	321	465488037
s1494	8	558	1416	6	25.24	36.12	1.43	6.35	7.36	1.16	334.8	485201908
s5378	35	1004	4584	179	16.23	49.26	3.04	4.23	13.64	3.22	319.8	179608417
s9234	36	2027	8396	211	17.19	26.25	1.53	5.23	8.27	1.58	901.3	516493137
s13207	31	2573	12593	669	28.62	32.12	1.12	8.72	9.2	1.06	1399.8	886879646
s35932	35	12204	30282	1728	56.73	73.49	1.3	18.82	24.47	1.3	2764.2	1506703528
s38584	12	11448	34498	1452	45.1	89.9	1.99	14.07	27.74	1.97	2779.8	3302189647
Average	-	-	-	-	-	-	1.52	-	-	1.56	-	-

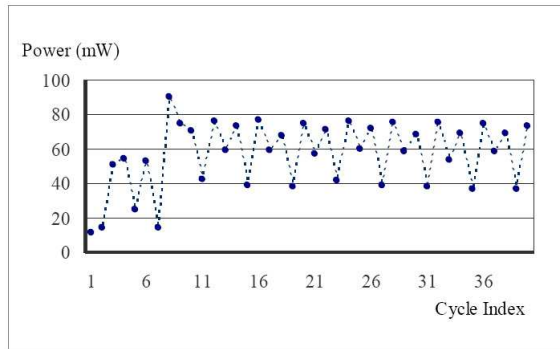


Fig. 10. Power value of each cycle for s38584.

TSMC 0.18 μ m and 0.13 μ m libraries, respectively.

A detailed power values of s38584 over the 40 cycles are shown in Fig. 10. The x-axis is the cycle index and the y-axis is the actual power value measured in mW. In Fig. 10, we find that the power values of even number cycles are higher than those of odd number cycles. This is because the ACO algorithm is only applied for the (V_{2n-1}, V_{2n}) cycle, but not for the (V_{2n-2}, V_{2n-1}) cycle. As a result, the power value is higher when V_{2n-1} switches to V_{2n} .

To compare the effectiveness of our ACO approach against the previous work [8], the experiments on ISCAS'85 benchmarks with the unit delay model are performed. The PSF value rather than the actual power value is reported in Table III. Column 2 lists the PSF value which represents the TD value

TABLE III
THE PSF COMPARISON OF THE GA METHOD [8] AND OUR ACO APPROACH.

Circuit	[8]		ACO		ACO / [8]
	PSF	Time (Sec.)	PSF	Time (Sec.)	
c432	1.175	7.2	1.155	10.84	0.983
c499	0.995	7.2	3.734	34.45	3.753
c880	0.823	13.2	0.976	24.92	1.186
c1355	0.883	13.3	1.064	18.03	1.205
c1908	1.838	34.3	1.070	29.38	0.582
c2670	1.464	89.9	2.065	161.06	1.411
c3540	1.460	74.1	1.109	85.07	0.759
c5315	1.714	170	1.486	288.25	0.867
c6288	7.359	617	32.300	782.26	4.389
c7552	2.020	311	2.404	290.23	1.190
Average	-	-	-	-	1.63

in [8]. Column 3 lists the CPU time in [8]. Columns 4 and 5 also list the PSF value and CPU time of our ACO approach. Column 6 lists the ratio of our approach and [8] on PSF value. Since the experimental setup and platforms are different, a direct value to value comparison is neither intended nor fair. It can be seen that, however, for more complicated circuits, such as c6288 (multiplier), our approach has a much higher PSF value. Thus, our ACO-based approach is a feasible alternative than GA-based approach for more complicated circuits. For ISCAS'89 sequential circuits, since the state reachability issue is not considered in [8], its PSF value will be overestimated. Thus, the comparison on sequential circuits are omitted here.

6. Conclusion

Getting tighter lower bound on the maximal power estimation requires the assistance of an efficient search algorithm in enormous search space. In this paper, an ACO algorithm for the maximal power estimation is proposed to avoid searching the whole solution space. The proposed ACO approach considers the delay model and the state reachability issue of sequential circuits. The experimental results show that in comparison with a random approach, the patterns generated by our approach result in much tighter lower bounds. As compared with the previous work, our results are also competitive.

REFERENCES

- [1] T. Chou and K. Roy, "Accurate Power Estimation of CMOS Sequential Circuits," *IEEE Trans. VLSI Systems*, pp. 369-380, Sep. 1996.
- [2] S. Chowdhury and J. S. Barkatullah, "Estimation of Maximum Current in MOS IC Logic Circuits," *IEEE. Trans. Computer-Aided Design*, pp. 642-54, Jun. 1990.
- [3] S. Devadas, K. Keutzer, and J. White, "Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation," *IEEE. Trans. Computer-Aided Design*, pp. 373-383, Mar. 1992.
- [4] M. Dorigo and T. Stützle, *Ant Colony Optimization*, Cambridge, MA: MIT Press, 2004.
- [5] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," in *Proc. Design Automation Conf.*, pp. 253-259, 1992.
- [6] A. Krstic and K. T. Cheng, "Vector Generation for Maximum Instantaneous Current through Supply Lines for CMOS Circuits," in *Proc. Design Automation Conf.*, pp. 383-388, 1997.
- [7] M. S. Hsiao, E. M. Rudnick, and J. H. Patel, "Effects of Delay Model in Peak Power Estimation of VLSI Sequential Circuits," in *Proc. Int. Conf. Computer-Aided Design*, pp. 45-51, 1997.
- [8] M. S. Hsiao, E. M. Rudnick, and J. H. Patel, "K2: An Estimator for Peak Sustainable Power of VLSI Circuits," in *Proc. Int. Symp. Low Power Electronics and Design*, pp. 178-183, 1997.
- [9] M. S. Hsiao, "Peak Power Estimation Using Genetic Spot Optimization for Large VLSI Circuits," in *Proc. Design, Automation, and Test in Europe*, pp. 175-179, 1999.
- [10] C. T. Hsieh, J. C. Lin, and S. C. Chang, "Vectorless Estimation of Maximum Instantaneous Current for Sequential Circuits," *IEEE. Trans. Computer-Aided Design*, pp. 2341-2352, Nov. 2006.
- [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," in *Proc. Design Automation Conf.*, pp. 23-28, 1995.
- [12] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Trans. VLSI Systems*, pp. 446-455, Dec. 1994.
- [13] F. N. Najm, S. Goel and I. N. Hajj, "Power Estimation in Sequential Circuits," in *Proc. Design Automation Conf.*, pp. 635-640, 1996.
- [14] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis," Technical Report UCB/ERL M92/41, Electronics Research Lab, Univ. of California, Berkeley, CA 94720, May 1992.
- [15] C. Small, "Shrinking Devices Put the Squeeze on System Packaging," *EDN*, pp. 41-46, Feb. 1994.
- [16] Synopsys Corporation. PrimePower Manual, May 2002.
- [17] T. Uchino, F. Minami, T. Mitsuhashi, and N. Goto, "Switching Activity Analysis Using Boolean Approximation Method," in *Proc. Int. Conf. Computer-Aided Design*, pp. 20-25, 1995.
- [18] H. J. M Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 468-473, 1984.
- [19] C. Wang, K. Roy, and T. Chou, "Maximum Power Estimation for Sequential Circuits Using a Test Generation-based Technique," in *Proc. Custom Integrated Circuits Conf.*, pp. 229-232, 1996.