ACO-based Peak Power Estimation in VLSI Circuits

Yi-Ling Liu Chun-Yao Wang
Dept. of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
g9562548@oz.nthu.edu.tw wcyao@cs.nthu.edu.tw

***Abstract*—Estimation of maximum power consumption is an essential task in VLSI circuit realization since power value significantly affects the reliability of the circuits. The key issue of this problem is which pattern pair causes 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, one new method with *Ant Colony Optimization* (*ACO*), which imitates the behavior of ants looking for foods, is proposed for peak power estimation. The approach returns the patterns which are highly suspicious to consuming the peak power. These generated patterns are then applied into a commercial power calculation tool, PrimePower, for actual power calculation under the TSMC 0.18 *μm* library. The gate delay issue and the valid state issue in sequential circuits are also considered in this paper. 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.**

I. 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 the reduction of the life-time of the chip, but also the failure rate of the components, which doubles for every $10°C$ increase in operating temperature [2]. In addition, the usage 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 delay, gate output capacitance, circuit architecture, 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 delay, gate output capacitance, 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 the accurate delay model since the behaviors of glitches and hazards are not taken into account under the zero-delay simulation.

One way to find 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] [4], they focus on average power. Average power is typically used as a reference for designing the power network in the ICs. This power value can be obtained from the signal switching probability. For peak power estimation, however, it needs to find out a specific pattern pair in the large space to get a tighter lower bound. This problem has been invested in [9], and is transformed into a weighted max-satisfiability problem, where a logic description is converted into a multiple-output Boolean function. 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.

Computing the maximum power cycles using symbolic transition counts is addressed in [10]. 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 maximum power consumption. Because of high complexity of this approach, it could only deal with small circuits, e.g., the test case reported in the paper contains less than 500 gates.

Another test generation-based technique is proposed in [3] to search the maximum transition nodes in the circuit. It tends to look 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 with the zero-delay model, which neglects the power contributed by glitches and hazards.

In this paper, the Ant Colony Optimization (ACO) [6] technique is applied for deriving the patterns with the PSF value in VLSI circuits. This method can deal with large combinational and sequential circuits with considering the general delays models The generated patterns are then applied into a commercial tool PrimePower [11] for the actual power calculation to demonstrate the effectiveness of our approach. The experimental results show that the maximum power consumption of the produced patterns is much higher than that of random patterns under the same amount of CPU time.

II. Preliminaries

A. Problem Formulation

The dissipated power in sequential circuits is govered by both initial state S_1 and input vectors V_1, V_2 as shown in Fig. 1 [8]. The initial state S_1 and input vector V_1 are used to initialize the internal nodes, and the state will change to S_2. After the application of new state S_2 and input vector V_2, some transitions that contribute to the power consumption occur in the internal nodes. The $2n$-cycle path that attempts to search the peak power dissipation from a valid initial state aims for finding one cycle that contributes the peak power consumption in the whole path.

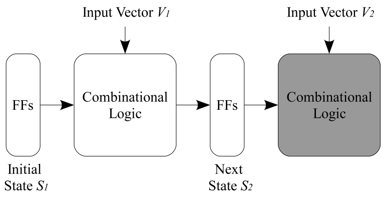

Fig. 1. Single cycle power model for sequential circuits.

For the computation of peak power consumption, consider a digital circuit with m internal nodes (gate outputs). During the time interval of $(-\frac{T}{2}, +\frac{T}{2}]$, where T is the cycle time, if the charging and discharging current is only considered, the average power dissipated in the time interval $(-\frac{T}{2}, +\frac{T}{2}]$ for all internal nodes can be computed as Equation (1) [5].

$$P = \frac{V_{dd}^2}{2} \sum_{i=1}^{m} C_i \frac{D_i}{T} \tag{1}$$

where V_{dd} is the supply voltage, and C_i is the gate output capacitance at node i; D_i is the number of transitions of one node switching from 0 to 1 or vice versa in the time interval $(-\frac{T}{2}, +\frac{T}{2}]$. Equation (1) can also be expressed as Equation (2).

$$P = \frac{V_{dd}^2}{2T} \sum_{i=1}^{m} C_i D_i \tag{2}$$

The Transition Density (TD) per node is detailed as Equation (3), and it is a parameter of ACO algorithm. The total number of capacitive nodes equals to the total number of gate inputs.

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

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

B. 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 strong pheromone concentrations. This mechanism of ants searching for food is known as Ant Colony Optimization (ACO) algorithm [6] and has been applied to solve many combinatorial optimization problems In general, ACO consists of three typical components that will be explained in Section IV.

C. Delay Model

Different cell delay models in the circuits lead to the disparity of power consumptions compared to the zero-delay model [7]. Thus, the general delay from the technology library is considered in this paper. But since the cell delay is process dependent and differs from a library to another library, the pattern set with peak power potential is enlarged to contain probabilistically a real peak power pattern. Finally, a power calculation tool, PrimePower, is used to derive the actual power values.

III. 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.

A. Initial Solution Construction

Initial solutions significantly affect the final results. To construct a better initial solution that has much more transition density, 10,000 random pairs of patterns (V_1, V_2) are applied to the circuits. After the simulation, the transition density of each pair of patterns can be calculated. This value indicates how this pair of patterns contributes to the maximum power consumption.

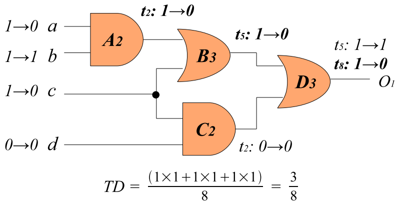

Fig. 2. Simulation process of the vector pair (1110, 0100).

In Fig. 2, assume the AND gates A and C are with 2-unit delay, and the OR gate B and D are with 3-unit delay. These delay values are also shown on the gates. When the input vector pair, $(1110, 0100)$, is simulated, some transitions occur in these gates, e.g., the gate A switches from 1 to 0 at time 2. At time 5, the gate B switches, but the gate D is intact. Subsequently, the transition density (TD) of this pair of vectors is derived as Fig. 2 shows. The numerator of TD is the summation of the transition number multiplying the number of gate outputs of each node. The denominator of TD is the number of capacitive nodes. The larger transition density, the more switching behavior and thus higher power consumption.

B. Pheromone Update

The pheromone is updated with respect to the quality of solution found. For each primary input i, each vector pair would cause four transition conditions 0→0, 0→1, 1→0, and 1→1. Four parameters $\tau^i_{0\to0}$, $\tau^i_{0\to1}$, $\tau^i_{1\to0}$, and $\tau^i_{1\to1}$, which represent the pheromones with respect to these four conditions of each primary input i are then generated. These parameters are used to accumulate the transition density of each input vector pair. The pheromone values also decrease with rate k in the mechanism of pheromone evaporation. It is expressed as Equation (4).

$$\tau^i_{0\to0}(t+1) = \triangle\tau^i_{0\to0}(t) + (1-k)\cdot\tau^i_{0\to0}(t) \tag{4}$$

where $k \in [0, 1]$ is the pheromone decay rate, and $\triangle\tau^i_{0\to0}(t)$ is the amount of pheromones added to the accumulated pheromone $\tau^i_{0\to0}$ at time t. The other three parameters are also expressed as Equation (4) except the transition condition.

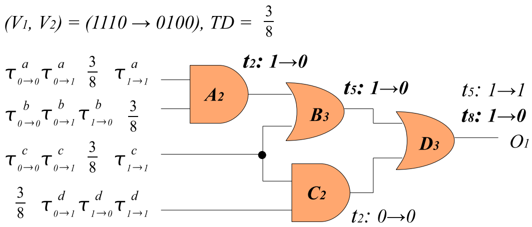
(V1, V2) = (1110 → 0100), TD = 3/8
τ^a_{0→0} τ^a_{0→1} 3/8 τ^a_{1→1}
τ^b_{0→0} τ^b_{0→1} τ^b_{1→0} 3/8
τ^c_{0→0} τ^c_{0→1} 3/8 τ^c_{1→1}
3/8 τ^d_{0→1} τ^d_{1→0} τ^d_{1→1}
t2: 1→0
A2
B3
C2
D3
t5: 1→0
t5: 1→1
t8: 1→0
O1
t2: 0→0

Fig. 3. The pheromone update process for the vector pair (1110, 0100).

Take Fig. 3 as an example that shows the same circuit as Fig. 2. For the primary input a, four parameters, $\tau^a_{0\to0}$, $\tau^a_{0\to1}$, $\tau^a_{1\to0}$, and $\tau^a_{1\to1}$, represent the pheromones of four transition conditions of one vector pair. These parameters are initialized to zero. After the input vector pair $(V_1, V_2) = (1110, 0100)$ is applied, the transition density of this vector pair on all inputs are derived. Then, according to the transition condition of each primary input, the derived transition density is taken for the indicator of the pheromone value needed to be updated. For instance, for the second primary input bit b, the transition condition between (V_1, V_2) changes from 1→1, and the value of parameter $\tau^b_{1\rightarrow 1}$ increases by $\frac{3}{8}$. For each input vector pair, the related pheromone values for each primary input are then updated continually.

C. Local Search

Ant Colony Optimization has been indicated to work well combined with the local search [6]. For this reason, the local search procedure is applied to the solutions of the maximum power consumption. After the application of 10,000 initial vector pairs, the top 100 input vector pairs with higher transition density are selected for local search. In addition, for each primary input, the highest and the secondary value of those four parameters can be obtained to compose two pair of input vector pairs that are in preparation for the crossover with those top 100 input vectors. The purpose of choosing the highest and secondary values is to improve the variety of vector pairs and the difference between these two values is sometimes very close.

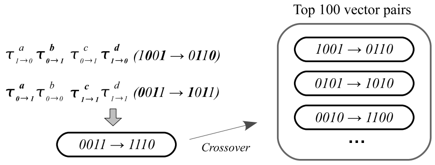
Top 100 vector pairs
$\tau^a_{1\rightarrow 0}\tau^b_{0\rightarrow 1}\ \tau^c_{0\rightarrow 1}\tau^d_{1\rightarrow 0}$ (1001 → 0110)
$\tau^a_{0\rightarrow 1}\tau^b_{0\rightarrow 0}\ \tau^c_{1\rightarrow 1}\tau^d_{1\rightarrow 1}$ (0011 → 1011)
0011 → 1110
Crossover
1001 → 0110
0101 → 1010
0010 → 1100
...

Fig. 4. Apply ACO algorithm for local search process.

As illustrated in Fig. 4, the left side is the two vector pairs that are composed of the highest and secondary parameter values from each primary input. The upper one consists of the highest parameter value from each primary input, and the lower one is the secondary parameter value. For instance, the four parameters of primary input b are $\tau^b_{0\rightarrow 0}$, $\tau^b_{0\rightarrow 1}$, $\tau^b_{1\rightarrow 0}$, and $\tau^b_{1\rightarrow 1}$, respectively. The highest value $\tau^b_{0\rightarrow 1}$ is then chosen for the purpose of composing the first vector pair. Other primary inputs also follow this procedure. The second vector pair consists of the secondary parameter values from each primary input. At the same time, the two vector pairs, $(1001 \rightarrow 0110)$ and $(0011 \rightarrow 1011)$ are then generated according to the corresponding transition condition. From these two pairs, for each primary input, either the highest or the secondary bit will be randomly chosen to combine one vector pair. In this figure, the vector pair $(0011 \rightarrow 1110)$ consists of the bit a and c of secondary vector pair and the bit b and d of the highest vector pair (marked in bold). In addition, the right side in Fig. 4 is the top 100 vector pairs from the 10,000 initial vector pairs. For each vector pair of the right side, one vector pair from the left side will be generated to accomplish the crossover. Then, one vector pair generated from the crossover will be applied to the circuit for simulation. After completing of 100 iterations, pheromone will be updated according to the transition density of each vector pair as previously mentioned. The local search will continue until the better solutions are found.

D. Sequential Circuits

The difference between combinational circuits and sequential circuits is the flip-flops in sequential circuits. The initial state of sequential circuits can be assigned to any arbitrary value only in fully-scanned circuits. If the initial state is assigned to an arbitrary value, the peak power estimation will be overestimated. Due to the reachability of initial states, this work starts from the reset state for sequential circuits, and aims to find one path that contain peak power consumption in some cycle. The process of searching this path is based on the architecture of combinational circuits.

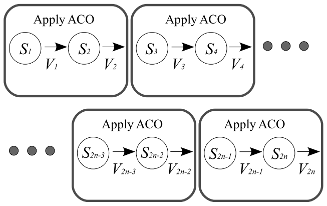

Fig. 5. Apply ACO algorithm for sequential circuits.

As illustrated in Fig. 5, the power consumption in one cycle can be calculated by deriving the vector (S_1, V_1, V_2). Since the S_1 starts from the reset state, the power consumption can be attained by observing the changes between (V_1, V_2). In the meanwhile, the ACO is also applied to search the vector pair (V_1, V_2) that consumes the maximum power. Then, the following cycles from (V_3, V_4) to (V_{2n-1}, V_{2n}) base on the same operations to form a path that will derive one of these $2n$ cycles that contributes peak power consumption. The ACO is only applied for (V_1, V_2), but not for (V_2, V_3), etc. Furthermore, every state of these $2n$ states is a valid state as every state is derived from the previous input vector.

IV. Experimental Results

In this section, the experimental results of the ACO algorithm for peak power estimation are addressed on a set of ISCAS'85 combinational benchmarks and ISCAS'89 sequential benchmarks. All of these benchmarks are synthesized under 0.18μm technology process, and the vector pairs produced by random and ACO approach combined with designs will also be applied to PrimePower for actual peak power consumption under 0.18 μm technology process. These experiments are conducted with C++ language with benchmarks of netlists level design description on a Sun Blade 2500 workstation with 4GB memory.

Table I demonstrates the experimental results of our approach and random generated patterns during the same time periods for ISCAS'85 combinational benchmarks. Column 1 lists the benchmarks and peak power consumption of all these benchmarks show tighter lower bound than random patterns. Column 2 lists the number of primary outputs in each benchmark, PI. Column 3 lists the number of gates in each benchmark, NG. Column 4 shows the number of capacitance nodes in each benchmark, NC. Column 5 and 6 lists the experimental results of random patterns and ACO approach under the same execution time, respectively. Column 7 lists the ratio of the maximum power estimation produced by ACO versus that reported by random patterns. The run time for these two approaches is listed in Column 8.

The number of vector pairs required for exhaustive simulation will exponentially grow with the primary input number. Take c432 for instance, the overall possible combinations of the two-tuple (V_1, V_2) are up to $4^{36} \simeq 4.7 \cdot 10^{23}$. However, it is impractical to simulate all of these pairs. For these reason, the ACO is applied to reduce the simulation time for this problem. In combination circuits, 10,000 random patterns are simulated for initial solution. Then, local search will repeat 32 iterations, and each iteration

TABLE I

PEAK POWER FOR ISCAS'85 COMBINATIONAL CIRCUITS.

Circuit	PI	NG	NC	RAN	ACO	$\frac{ACO}{RAN}$	Time(m)
c432	36	204	343	10.9	22.3	2.05	0.97
c499	41	276	440	10.48	17.9	1.71	0.56
c880	60	470	755	37.2	71.7	1.93	0.76
c1355	41	620	1096	27.9	29.45	1.06	2.07
c1908	33	939	1523	15.27	18.47	1.21	2.22
c2670	234	1567	2216	114.8	241	2.1	5.35
c3540	50	1742	2961	65.1	237.1	3.65	5.58
c5315	178	2609	4509	131.4	172.3	1.31	5.33
c6288	32	2481	4832	204.6	233.4	1.14	15.53
c7552	207	3828	6252	240.8	338.8	1.41	23.33
Average						1.76	

TABLE II

PEAK POWER FOR ISCAS'89 SEQUENTIAL CIRCUITS.

Circuit	PI	NG	NC	FFs	RAN	ACO	$\frac{ACO}{RAN}$	Time(m)
s344	9	101	295	15	9.46	11.95	1.26	0.97
s386	7	118	359	6	2.97	2.97	1.0	0.56
s1196	14	388	1045	18	23.4	39.72	1.7	2.07
s1423	17	490	1300	74	13.29	18.47	1.39	2.22
s1488	8	550	1410	6	17.18	25.73	1.5	5.35
s1494	8	558	1416	6	25.24	36.12	1.43	5.58
s5378	35	1004	4584	179	16.23	49.26	3.04	8.76
s9234	36	2027	8396	211	17.19	26.25	1.53	15.35
s13207	31	2573	12593	669	28.62	32.12	1.12	23.33
s35932	35	12204	30282	1728	56.73	73.49	1.3	46.07
s38584	12	11448	34498	1452	45.1	89.9	1.99	46.33
Average							1.52	

contains 100 vector pairs with pheromone evaporation rate of 0.1. To avoid being trapped in local space, some other 10,000 random pairs will be applied for the variety of vector pairs. At last, the highest 100 vector pairs will then be outputted as the testbench for PrimePower. According to the power analysis from the tool, these higher vector pairs also contribute average higher power consumption than random pattern pairs do. This tool also indicates the highest power consumption within these 100 pattern pairs. Under the same simulation periods, on the average, the ACO approach gives 76% tighter lower bound on maximum power dissipation than the bound obtained with random pattern generator. The total vector pairs of ACO required for simulation are approximately 200,000 while random pattern pairs are over 1 million under the same time periods.

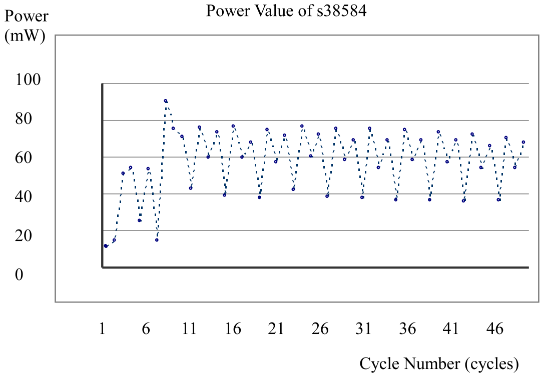

Fig. 6. Power dissipation of each cycle for s38584.

Table II displays the experimental results of our approach and random generated patterns during the same time periods for ISCAS'89 sequential benchmarks. Column 5 lists the number of flip-flops within each circuit. Because of reachability factor, ACO for sequential circuit attempts to search one path that contributes maximum power within some cycle. The power consumption over each cycle of s38584 is reported in Fig. 6. The x-axis is the cycle numbers and the y-axis is the power values in mW. After the application for some cycles, the power consumption tends to the regular behavior that the power values within the even cycles are higher than those of odd cycles. This is because the ACO is only applied for $(V_{2n\text{-}1}, V_{2n})$ but not for $(V_{2n\text{-}2}, V_{2n\text{-}1})$. As a result, the power value is higher when $V_{2n\text{-}1}$ switches to V_{2n} For this regularity, the n is set to be any positive integer from 1 to 20. Finally, the search path of power consumption for sequential circuits will then be outputted as the testbench for PrimePower. The tool will report the maximum power consumption within some cycle of these cycles. Under the same simulation period, on the average, the ACO approach gives 52% tighter lower bound on maximum power dissipation than the bound obtained with random pattern generator.

V. CONCLUSIONS

Getting tighter lower bounds on maximum power requires efficient search algorithm in enormous search space. In this paper, the ACO algorithm for the maximum power estimation is proposed to avoid searching the overall solution space. The proposed heuristics have been shown to be very effective in large combinational and sequential circuits. The experimental results reveal that in comparison with a random vector generator, the patterns generated applying our approach result in much tighter lower bounds up to an average of 76% for ISCAS'85 and 52% for ISCAS'89 on maximum power dissipation.

REFERENCES

[1] 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.

[2] C. Small, "Shrinking Devices Put the Squeeze on System Packaging," *EDN.*, pp. 41-46, 1994.

[3] 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.*, 1996.

[4] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Trans. VLSI Systems*, vol. 2, no. 4, pp. 446-455, 1994.

[5] 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.

[6] M. Dorigo and T. Stützle, Ant Colony Optimization Cambridge, MA: MIT Press, 2004.

[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, "Peak Power Estimation Using Genetic Spot Optimization for Large VLSI Circuits," in *Proc. of the IEEE Design, Automation, and Test in Europe Conference*, pp. 175-179, 1999.

[9] S. Devadas, K. Keutzer, and J. White, "Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation," in *IEEE. Trans. Computer-Aided Design*, pp. 373-383, 1992.

[10] 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.

[11] Synopsys Corporation. PrimePower Manual, May 2002.