A Co-evolutionary Algorithm for Dynamic Power Minimization During Scan Testing

Alodeep Sanyal†
Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003
Email: asanyal@ecs.umass.edu

Artem Sokolov, Yashwant Malaiya and Darrell Whitley
Computer Science Department
Colorado State University
Fort Collins, CO 80523
Email: {sokolov, malaiya, whitley}@cs.colostate.edu

***Abstract*— Power dissipation during scan testing is becoming an important concern as design sizes and gate densities increase. In scan-based testing, a significant fraction of total test power is dissipated due to switching activity in the combinational logic block as well as flip-flops. In this paper, we show that the switching activity in the circuit nodes during testing is a function of both test vector ordering and flip-flop ordering. We present a co-evolutionary algorithm which minimizes the total switching activity during testing by jointly optimizing the test vector order and the flip-flop order. Experimental results on ISCAS89 benchmark circuits show that our solution consistently outperforms one of the current state-of-the-art method.**

Keywords: sequential circuit, controllability, observability, scan design, automatic test pattern generator (ATPG), NP-hard problem, evolutionary and co-evolutionary algorithms, directed graph, path traversal problem, greedy heuristic, simulated annealing, traveling salesman problem, local search, fault coverage

I. Introduction

A. Motivation

The growing size of very large scale integration (VLSI) circuits, high transistor density, and popularity of low-power hand-held devices are making minimization of power dissipation an important issue in VLSI design. The amount of heat generated limits the density of a chip. Reduction of power dissipation also permits the use of smaller package size. This can reduce weight of portable products and prolong battery life.

Power dissipation during test application also plays a key role. Circuits are often designed to operate in two modes: *normal mode* and *test mode*. Systems registers, when activated during test mode, can be in states that are not reached in normal mode. As a result, state transitions that are not possible during normal mode are often possible during test mode and can lead to significantly higher switching activity in the circuit nodes causing a substantial drop in the power supply voltage. Excessive power dissipation during testing could damage the chip, or prevent periodic testing of such equipment. In the case of multi-chip modules, it has been observed that the potential advantages in circuit density and performance of the technology cannot be realized without access to fully tested, unpackaged integrated circuits or what are called *bare die*. Absence of packaging precludes the use of traditional heat removal techniques during the bare die testing. In such cases, power dissipated during testing can adversely affect the overall yield, thus adding to the production cost [1]. Moreover, during *wafer testing* the number of power pad pins accessible is significantly less compared to those accessible during normal operation mode [2]. Drastic increase in leakage power with newer technology generations further limit the maximum dynamic power that can be consumed during testing.

†This research was carried out when the author was with the Computer Science Department of Colorado State University, Fort Collins, CO 80523

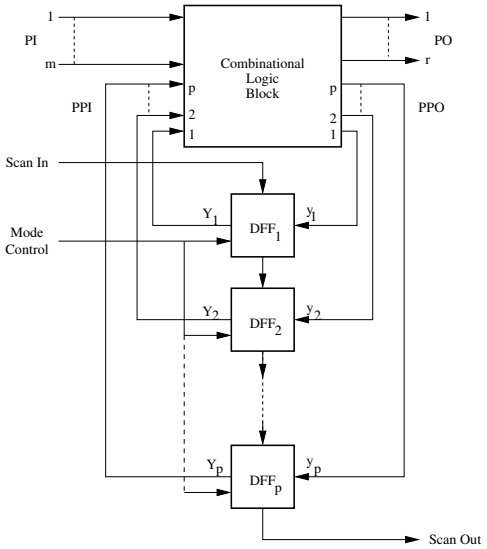

Fig. 1. Mealy machine representation of a prototype sequential circuit

B. Sequential circuits and scan-based testing

A sequential circuit, when represented as a *Mealy machine* [3], is essentially composed of a combinational logic block C and a finite set of memory elements (such as flip-flops) as shown in Figure 1. Testing a sequential circuit is characteristically different from testing a combinational circuit in the sense that for the sequential circuits the primary outputs (PO) are function of not only the primary inputs (PI) but also the state stored in the flip-flops. Therefore, to test for a specific value in a primary output, an appropriate state has to be latched in the flip-flops before the application of a specific vector in the primary inputs (PI). The outputs of the flip-flops, which become inputs to C are sometimes called *pseudo-primary inputs* (PPI) and the inputs to the flip-flops, which come out of C are likewise called *pseudo-primary outputs* (PPO).

It has been observed that internal nodes of a complex sequential circuit are often difficult to control or observe. In order to improve the *controllability* and *observability* [4] of the internal nodes several techniques have been proposed of which *scan design* has become most popular.

In a typical scan design, all the flip-flops of the sequential circuit are connected in a particular order to form a chain (Figure 1). Before

a test is applied to the PIs, an appropriate state is latched in the flip-flops by serially scanning in the logic values. After the test, the new state is stored in the flip-flops and then scanned out serially. If tests are performed in sequence, the scan-out and the scan-in operations can be overlapped to make it more efficient [4].

A scan circuit operates in two modes controlled by a *mode control* signal: i⟩ *scan mode*, and ii⟩ *functional/test mode*. If there are p flip-flops in the scan chain, it takes p clock cycles to scan in a particular state in the scan mode, followed by the application of the test in one clock cycle in test mode.

During the scan mode of operation, switching activity in the circuit becomes fairly high because of switching in the scan chain, in the clock tree and in the combinational logic. It has been reported that power dissipated during test application can be significantly higher (sometimes, 100-200%) than that during the circuit's normal operation [5].

C. Organization of the paper

Prior experimental results show that test vector reordering [1], [6]-[9] in the context of combinational circuit testing, and test vector as well as scan cell reordering [1], [10] in the context of scan-based sequential circuit testing can reduce power dissipation by an order of magnitude compared to the original unordered test set generated by a traditional automatic test pattern generator (ATPG) and applied to a randomly ordered scan-based testing environment. Both test vector reordering and scan cell reordering separately fall into the class of *NP-hard* problems [1], [11].

In this paper, we first state the problem of joint reordering of the test set and scan cells. We then make use of *co-evolutionary search* [12] to minimize the total switching activity in the combinational logic block as well as in the scan cells during testing. We show that our solution consistently outperforms a state-of-the-art method that has been previously applied to the problem [1].

The organization is as follows: in Section II, we briefly review the related work in this field. Section III establishes the model for dynamic power dissipation during scan testing. In Section IV, we provide a precise mathematical formulation of the problem and illustrate our proposed solution in Section V. We analyze the experimental results for our proposed method in Section VI. We conclude in Section VII with a brief discussion on future directions.

II. Related work

Power dissipation issues are addressed at various stages of circuit design. For example, circuit synthesis to reduce the average switching activity is described in [13]-[15], technology mapping that targets low power dissipation is considered in [16], [17], and physical design for low power is considered in [18].

In the domain of circuit testing, low-power dissipation test methods have been investigated thoroughly for combinational and sequential circuits. Dabholkar *et al.* [1] have proposed several heuristics both for combinational circuits and scan-based sequential circuits. They show that test vector ordering and scan cell ordering individually fall into the NP-hard class. The authors construct a *directed graph* with edge and node weights representing the dynamic power dissipation in the combinational logic and the flip-flops during the scan and test cycles, which reduces the given problem to a *path traversal problem*. Then they propose a $O(n^2 logn)$ *greedy heuristic* for the test vector ordering problem where n is the number of test vectors, and a *simulated annealing heuristic* for the combined test vector and scan cell ordering problem. The idea of scan cell ordering has also been studied by Ghosh *et al.* [10], Bonhomme *et al.* [19], and, in the context of reducing test application time by Narayanan *et al.* [20].

Several other methods aiming to solve power-related problems associated with scan-based test have been proposed recently, which fall into the following four broad categories:

Low transition test patterns: [21]-[23]. These methods reduce the number of transitions in the scan-in vectors, and consequently the shift-power component caused by scan-in transitions. However, these methods have no direct control over the number of transitions in the scan-out vectors, thus, overall reduction in power cannot be guaranteed.

Power conscious ATPG algorithms: [24]-[27]. These are special ATPG algorithms which aim to decrease the number of transitions in scan-in and scan-out vectors for shift-power reduction, and also to decrease the Hamming distance between test stimulus vectors and the corresponding test response vectors for capture cycle-power reduction. However, the test sets generated by these ATPG algorithms are generally much larger compared to test sets generated with regular ATPG algorithms.

Special scan cells: [28]-[31]. The approach proposed in [28] inserts blocking logic on the outputs of the scan cells in order to block the shift ripple at the inputs of the circuit. Although this method substantially reduces power dissipation during the shift cycle by completely eliminating the useless switching activity in the combinational logic block, it introduces undesired delay on the data path due to the blocking logic which has a negative impact on circuit's performance. The work presented in [26] improves the solution from [28] by inserting blocking logic only on the outputs of a limited number of flip-flops which are not on critical paths. Other variants to improve this method [28] have been proposed by Huang *et al.* [29], Parimi *et al.* [30], and Bhunia *et al.* [31].

Scan chain partitioning: [32]-[35]. Several scan chain partitioning approaches have also been reported in literature. These design approaches either focus on minimizing the dynamic power dissipation ([32]-[34]), or on peak power minimization [35] during scan testing.

In a previous paper we studied the problem of dynamic power minimization during combinational circuit testing viewed as a *Traveling Salesman Problem* (TSP) [6]. We explored application of *local search* (*viz.* 2-opt heuristic) and *evolutionary algorithms* to test set reordering and performed quantitative comparison with previously used deterministic techniques. We also considered reduction of the original test set framed as a dual-objective optimization problem with switching activity and *fault coverage* being two objective functions. We reported favorable results for our proposed solutions.

III. Power dissipation model

The two components of power dissipated in a CMOS circuit [36] are i) *static dissipation* due to leakage current through the channel and the tunneling current through the gate oxide drawn continuously from the power supply (P_{st}), and ii) *dynamic dissipation* due to switching transient current (P_{sc}) and charging and discharging of load capacitances (P_d). The total power dissipation (P_{total}) is given by

$$P_{total} = P_{st} + P_{sc} + P_d \quad (1)$$

CMOS leakage current has been steadily rising due to increased sub-threshold and gate oxide leakages and has gained attention in the literature [37]. However, power due to switching activity in the circuit nodes still dominates overall power consumption, and is only considered here.

P_d is the power required to charge and discharge the output capacitance load of every gate, both in combinational logic block and in the flip-flops. P_d is approximated as follows [1]:

$$P_d = 1/2 \times C \times V_{DD}^2 \times N_G \times f \quad (2)$$

where C is the output capacitance, V_{DD} is the supply voltage, N_G is the total number of gate output transitions ($1 \rightarrow 0$ and $0 \rightarrow 1$), and f is the clock frequency.

Equation (2) implies that power is dissipated at a node when the input vector is changed from T_i to T_{i+1}. Let $P_C(T_i, T_{i+1})$ be the total power dissipated in a combinational circuit C when inputs change from T_i to T_{i+1}. Then

$$P_C(T_i, T_{i+1}) = \sum_{j \in Set of Nodes} 1/2 \times C_j \times V_{DD}^2 \times N_{G_j} \times f \quad (3)$$

Thus, power dissipated at a node is proportional to the number of transitions (N_G) at that node. This depends on the gate delays. Under the zero-delay model, all gates are assumed to have zero delay. The four possible transitions and the corresponding logic levels in this model are:

- *static zero*: logic level remains zero
- *static one*: logic level remains one
- *rising*: logic level changes from zero to one
- *falling*: logic level changes from one to zero

The alternative is the general-delay model where gates can have arbitrary delays, and, therefore, takes the glitches into account. In this paper, we have assumed the zero-delay model.

IV. Dynamic power dissipation during scan testing

Under the assumption of zero delay model, we now focus on analyzing different components of dynamic power dissipated in a scan circuit during the scan and test mode of operations.

1) **Scan mode**: During the serial scan-in operation, dynamic power is consumed in the following two ways:
   i⟩ *Switching activity in the combinational logic block:* When the *pseudo primary inputs* (PPI) are serially scanned in, some useless switching activity takes place in the combinational logic block because of repeated bit flips in the memory elements, which causes a significant amount of dynamic power consumption. This can be prevented by any of the toggle suppression techniques studied in [28]-[31].
   ii⟩ *Switching activity in the flip-flops:* In the scan testing environment when the tests are performed in sequence, the scan-out of the output state of the current test vector and the scan-in of the input state of the next test vector are overlapped to make it more efficient [4]. In order to compute the total switching in the flip-flops during the scan operation we follow the following procedure:
   - First, concatenate scan-out state of the current test vector with the scan-in state of the next test vector.
   - Second, traverse the resulted string and count the transitions from 0 to 1 and from 1 to 0.
   - Third, multiply the transitions by (number of flip-flops - position of the transition from the concatenation point). This will account for the number of flip-flops that the transition passes through. For example, the concatenation point passes through all flip-flops and must be counted the appropriate number of times.
   - Finally, sum up all the weighted results to get the total switching activity.
2) **Test mode**: During the application of test, dynamic power is consumed in the following two ways:
   i⟩ *Switching activity in the combinational logic block:* When a test vector is applied in the primary inputs (PIs) along with the present state applied in the pseudo primary inputs (PPIs), dynamic power is dissipated in all the nodes of the combinational logic block C, which change from the logic value 0 to 1, or vice versa. Equation (3) implies the total dynamic power dissipation in C when the test vector is changed from T_i to T_{i+1}.
   ii⟩ *Switching activity in the flip-flops:* At the end of the test cycle, a new state is latched in the flip-flops replacing the old state scanned in before. It involves switching activity in the flip-flops. However, this switching will always take place irrespective of the order of the flip-flops in the scan chain. By observing the orthogonality of this switching activity in the context of test vector and scan cell ordering problem, we do not consider it in our optimization criterion.

Therefore, the total dynamic power dissipation during scan testing, P_{total}, can be expressed as follows:

$$P_{total} = P_{F,S} + P_{C,T} \quad (4)$$

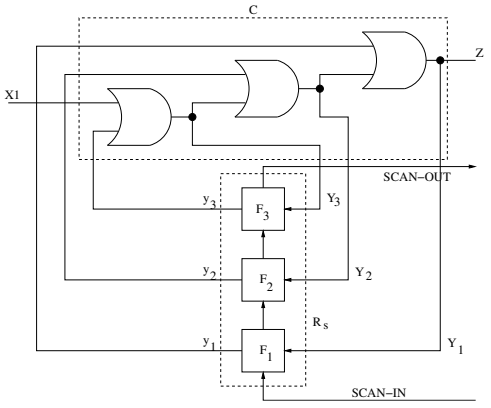

Fig. 2. An example illustrating factors affecting power dissipation

Cycle	F1	F2	F3	X1	R_s	C
0	1	0	0	0	-	-
1	1	1	0	0	10	2
2	0	1	1	0	14	1
3	1	0	1	0	14	0
4	1	1	1	0	10	0

TABLE I

NUMBER OF TRANSITIONS IN R_s AND C ON THE APPLICATION OF TEST VECTOR PAIR $\langle T_1, T_2 \rangle$

where,

$P_{F,S}$ = dynamic power dissipated in the flip-flops in scan mode, and

$P_{C,T}$ = dynamic power dissipated in the combinational block in test mode.

V. Problem definition

We reproduce the example cited in [1] to illustrate the factors affecting dynamic power dissipation during scan testing and to establish the point that number of transitions during scan is a function of both vector order and scan cell order.

Figure 2 refers to a simple sequential circuit having three OR gates and three flip-flops. Let X_1 be its primary input, $F = \{F_1, F_2, F_3\}$ be the scan flip-flops, y_i be the present state, and Y_i denote the next state for flip-flop F_i. An input vector of C is an ordered tuple of values $\langle x_1, f_1, f_2, f_3 \rangle$. We assume zero-delay model and the capacitance of each node to be equal to its fanout.

Let T_1 = ⟨0100⟩, T_2 = ⟨0111⟩; and let the flip-flop order be $\langle F_1, F_2, F_3 \rangle$. Tables I and II show the number of transitions in R_s and C in every cycle during the application of the test vector pairs $\langle T_1, T_2 \rangle$ and $\langle T_2, T_1 \rangle$, respectively. When the vector order is

Cycle	F1	F2	F3	X1	R_s	C
0	1	1	1	0	-	-
1	0	1	1	0	10	0
2	0	0	1	0	10	0
3	1	0	0	0	14	2
4	1	0	0	0	0	0

TABLE II

NUMBER OF TRANSITIONS IN R_s AND C ON THE APPLICATION OF TEST VECTOR PAIR $\langle T_2, T_1 \rangle$

Cycle	F1	F3	F2	X1	R_s	C
0	1	0	0	0	-	-
1	0	1	0	0	14	4
2	1	0	1	0	18	2
3	1	1	0	0	14	2
4	1	1	1	0	10	0

TABLE III

NUMBER OF TRANSITIONS IN R_s AND C ON THE APPLICATION OF TEST VECTOR PAIR $\langle T_1, T_2 \rangle$ WITH A DIFFERENT SCAN CHAIN ORDER $\langle F1, F3, F2 \rangle$

$\langle T_1, T_2 \rangle$, the total number of transitions (i.e., the sum of all the transitions in columns R_s and C) is 51, while for $\langle T_2, T_1 \rangle$, it is 36. Thus, the total number of transitions during scan is a function of vector order. If we change the flip-flop order to $\langle F_1, F_3, F_2 \rangle$, Table III shows that the number of transitions during $\langle T_1, T_2 \rangle$ is 64, whereas for the same vector order with flip-flop order $\langle F_1, F_2, F_3 \rangle$ was 51. Thus, the total number of transitions during scan is also a function of flip-flop order.

With the above example illustrating the dependence of switching activity during scan testing on test vector as well as flip-flop ordering, we now formally define our problem as follows:

Problem statement: Given a sequential circuit Q with combinational part C, flip-flop set $F = \{f_1, ..., f_m\}$, and test set $T = \{t^1, ..., t^l\}$, find a flip-flop ordering $R = \langle s^1, ..., s^m \rangle$ of F and test vector ordering $S = \langle s_1, ..., s_l \rangle$ of R such that P_{total}, defined in Equation (4), is minimized.

VI. THE PROPOSED APPROACH

A. Common Objective

Minimization of the total switching activity can be performed by either reordering the test vectors, or reordering the flip-flops. While these are two distinct minimization problems, they share the common objective function. More specifically, reordering test vectors affects the switching activity during both scan and test modes, while reordering flip-flops affects only the former. This is because of the fact that only during scan in and scan out operations, the relative ordering of scan flip-flops influences the switching activity in the flip-flops. However, once a state is latched in the scan chain followed by application of the test, the flip-flops will store the next state (possibly causing switching activity) independent of the relative order of the flip-flops in the chain. Due to this overlap, locally optimal solution found by reordering the test vectors can be further improved by reordering the flip-flops. Of course, the improved solution may no longer be locally optimal with respect to the test vector ordering, and further improvement may still be possible. By alternating between the two optimization problems, we aim to drive the total switching activity to some value that is locally optimal to both, the ordering of test vectors and the ordering of the flip-flops.

Both problems can be easily expressed as search, first through the space of test vector permutations, then through the space of flip-flop permutations. We consider two methods here that are used to perform the actual search. First one is simulated annealing, a method that has already been successfully applied to dynamic power minimization in scan circuits [1]. The other approach is genetic algorithms. Both methods used permutations to decode solutions. Following are more detailed descriptions of the two.

B. Simulated annealing

Simulated annealing [38] is a form of local search that allows acceptance of worse moves. The probability with which a worse move is accepted is dependent upon how much worse that move is, and also decreases with time according to some "cooling" schedule. The implementation used for this paper was the following:

1) Construct a random initial solution S
2) Select an initial temperature T
3) **for**($iter$ = 1 to N) repeat steps 4-9
4) **for**(i = 1 to L) repeat steps 5-8
5) Pick a random neighbor S' of S.
6) Let Δ = (cost(S') - cost(S)) / cost(S)
7) **if**($\Delta \leq 0$) $S \leftarrow S'$
8) **else** $S \leftarrow S'$ with probability $e^{-\frac{\Delta}{T}}$
9) $T \leftarrow rT$

The major difference of this implementation to that proposed by Dabholkar *et al.* [1] is in the normalization of the difference term Δ. This was done to ensure consistent behavior of the algorithm as iterations progressed. All parameters were chosen to be consistent with [1]. The time complexity of the above algorithm is $O(NL)$. That is, L was fixed at 20, the initial temperature was chosen to be 1.5, and random neighbors were generated by swapping two (randomly sampled without replacement) elements in a permutation. We chose the number of iterations N to be 150, to give a total of $20 * 150 = 3000$ objective function evaluations. The cooling variable r was fixed at 0.96 to yield 5% probability of accepting a move that was 1% worse than the current solution towards the end of the search.

C. Genetic algorithms

Genetic algorithms are outlined by Whitley [39]. The general idea behind this approach is to maintain a set of solutions and to recombine pairs of these solutions to produce better ones. We used an elitist generational framework in this paper:

1) Create an initial population P.
2) **for**(i = 1 to $nGen$) repeat steps 3-11
3) Create a new empty population P'
4) Add the current best solution to P'
5) **for**(j = 1 to $pop\_size/2$) repeat steps 6-9
6) Select two solutions from the current population using the selection scheme
7) Apply crossover operator to produce two new solutions
8) Apply mutation to the new solutions
9) Add the new solutions to P'
10) $P \leftarrow P'$
11) Truncate the population to its original size

We used random tournament selection with selective pressure of 2.0 [40], weight-biased edge crossover [41], and mutation rate of 0.5% for parameter choices. The mutation operator simply samples the appropriate portion of elements in a permutation without replacement and moves the sampled elements to the end of the permutation. The population size pop_size and the number of generations $nGen$ were chosen to be 100 and 30 respectively. This again yields $30 * 100 = 3000$ objective function evaluations.

D. Co-evolutionary search

After discussing each search technique separately, we can put all ideas together and construct the following implementation:

1) Randomly choose test vector ordering T_{cur}
2) Randomly choose flip-flop ordering D_{cur}.
3) Set C = sw-scan(T_{cur}, D_{cur}) + sw-test(T_{cur}, D_{cur})
4) **for**(i = 1 to $nIter$) repeat steps 5-12
5) Fix T_{cur}.
6) $D_{new} \leftarrow$ search(sw-scan).
7) $C_{new} \leftarrow$ sw-scan(T_{cur}, D_{new}) + sw-test(T_{cur}, D_{new})
8) **if**($C_{new} < C$) $D_{cur} \leftarrow D_{new}$ and $C \leftarrow C_{new}$
9) Fix D_{cur}.
10) $T_{new} \leftarrow$ search(sw-test).
11) $C_{new} \leftarrow$ sw-scan(T_{new}, D_{cur}) + sw-test(T_{new}, D_{cur})
12) **if**($C_{new} < C$) $T_{cur} \leftarrow T_{new}$ and $C \leftarrow C_{new}$

	Unoptimized	Simulated Annealing		Genetic Algorithms	
	Sw. Act.	Sw. Act.	% Imp.	Sw. Act	% Imp.
s1196	42854 ± 1100	37196 ± 1012	13.2	28439 ± 749	33.6
s1238	43608 ± 915	37769 ± 1100	13.4	29585 ± 649	32.2
s1488	26746 ± 808	23139 ± 721	13.5	11209 ± 246	58.1
s1494	26204 ± 774	22408 ± 776	14.5	11032 ± 208	57.9
s298	4331 ± 237	3430 ± 250	20.6	3189 ± 179	26.4
s344	3708 ± 291	2698 ± 187	27.2	2600 ± 193	29.9
s349	3767 ± 244	2743 ± 210	27.2	2613 ± 198	30.6
s382	8604 ± 497	6704 ± 438	22.1	6111 ± 414	29.0
s386	4075 ± 140	3182 ± 94	21.9	2180 ± 50	46.5
s444	8314 ± 436	6751 ± 444	18.8	6247 ± 379	24.9
s510	4492 ± 155	3613 ± 127	19.6	2620 ± 81	41.7
s526	16377 ± 673	13940 ± 597	14.9	12202 ± 520	25.5
s5378	3884359 ± 108383	3856102 ± 90837	0.7	2813060 ± 78113	27.6
s641	17279 ± 1275	13120 ± 818	24.1	10933 ± 698	36.7
s713	17441 ± 883	12888 ± 840	26.1	11066 ± 533	36.6
s820	10516 ± 407	8899 ± 306	15.4	5274 ± 141	49.8
s832	10139 ± 360	8706 ± 256	14.1	5144 ± 117	49.3
s953	37314 ± 1182	29734 ± 871	20.3	25841 ± 928	30.7

TABLE IV

SWITCHING ACTIVITY FOR TEST VECTOR AND FLIP-FLOP ORDERINGS. FIRST COLUMN PRESENTS THE UNORDERED CASE, AS GENERATED BY ATALANTA. THE OTHER TWO COLUMNS DEMONSTRATE THE SWITCHING ACTIVITY AFTER THE ORDERINGS HAVE BEEN OPTIMIZED, USING EITHER SIMULATED ANNEALING OR GENETIC ALGORITHMS. THE RESULTS ARE PRESENTED IN THE (MEAN ± ST.D.) FORM. THE STATISTICS THEMSELVES HAVE BEEN COMPUTED ACROSS 30 RUNS. PERCENT IMPROVEMENT IS SPECIFIED RELATIVE TO THE UNSORTED COLUMN.

The meaning of functions is as follows. `sw-scan` computes switching activity during the scan mode, given the current orderings of test vectors and flip-flops. Function `sw-test` has a similar meaning, but in the context of the test mode. Function `search` is either simulated annealing or genetic algorithm and its argument is the objective function used for the search. The time complexity of the above co-evolutionary algorithm is $O(nGen \times pop_size \times nIter)$.

Because reordering of test vectors affects the switching activity during both *scan* and *test* modes, step 10 may actually result in an increase of switching activity returned by `sw-scan`. The change will, nevertheless, be accepted in step 12 if the total switching activity is lower. On the other hand, step 6 does not result in a similar issue, since switching activity of the test mode is unaffected by the order of flip-flops.

For all our experiments, the number of iterations $nIter$ was fixed at 25, resulting in a total of 50 calls to either search method. The resulting number of objective function evaluations was around $3,000 * 50 = 150,000$.

VII. EXPERIMENTAL RESULTS

We compare the performance of genetic algorithms and simulated annealing in the context of co-evolutionary search. We chose simulated annealing for comparison because of its favorable performance in minimization of switching activity during scan testing compared to other search algorithms, as was demonstrated by Dabholkar *et al.* [1].

The experimental results are outlined in Table IV. We performed 30 separate runs where we first generated test vectors using ATALANTA [42], then used co-evolutionary search with simulated annealing and genetic algorithms to optimize the switching activity. Because ATALANTA generates a different set of test vectors at each run, the total switching activity at the initial unoptimized step has some variance. This variance is presented in the first column of Table IV.

Switching activity in the final solution found by simulated annealing is presented in the second column. The improvement achieved with this technique is in the range of 10-25%, which is consistent with results presented by Dabholkar *et al.* [1]. Similarly, objective function values for solutions found by genetic algorithms are outlined in the last column. All statistics are presented in the (*mean* ± *st.d*) form.

Note that the scan cell reordering technique should be applied before the synthesis process, while test-vector ordering can be applied any time before test application. However, since the proposed solution relies on alternating between optimization of the two its application must be performed prior to the synthesis of the circuit. Since this is not a real-time operation, run times are not relevant and, therefore, not reported.

Most algorithms are sensitive to the choice of parameters, but tuning parameters is of most interest primarily to the end-user. We have taken the best result from a search algorithm applied to this problem, namely simulated annealing. We have then replicated that result, choosing the same set of parameters for the algorithm as was proposed in the original paper, and showed that further improvement is possible. The results don't demonstrate that a GA with the best set of parameters will outperform SA with its best set of parameters, but it does demonstrate that search algorithms applied to this problem so far can be improved upon. Due to the choise of parameters, comparison to state of the art is also difficult. The issue is further complicated by the fact that there are no standards in computing the switching activity.

VIII. CONCLUSIONS AND FUTURE WORK

The contribution of this paper lies in examining the problems of test vector and scan cell reordering as two distinct problems that share a common objective function. Alternating between optimizing each ordering, we were able to drive the overall switching activity during scan testing to a locally optimal value that was significantly better than the initial case generated by ATALANTA [42]. We also demonstrated that genetic algorithms yield imporved performance when compared to the best result from search algorithms applied to the problem of switching activity minimization during scan testing.

In this paper, we did not explore area overhead due to scan cell ordering. It would be interesting to study a dual optimization problem where switching activity and chip area overhead would be two objective functions in the context of scan cell ordering.

REFERENCES

[1] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. M. Reddy. Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits During Test Application. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 17, No. 12, December 1998, pp. 1325-1333*

[2] S. Kundu, T. M. Mak, and R. Galivanche. Trends in manufacturing test methods and their implications. *International Test Conference, 2004*

[3] Z. Kohavi. Switching and Finite Automata Theory. *New York: McGraw-Hill, 1978*

[4] M. L. Bushnell, and V. D. Agrawal. Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits. *Spinger Academic Publications, New York, 2000*

[5] Y. Zorian. A Distributed BIST Control Scheme for Complex VLSI Dvices. *VLSI Test Symposium, pp. 4-9, 1993*

[6] Self reference

[7] S. Chattopadhyay and N. Choudhary. Genetic Algorithm based Approach for Low Power Combinational Circuit Testing. *16th IEEE International Conference on VLSI Design, pp. 552-557, January 4-8, 2003*

[8] P. Flores, J. Costa, H. Neto, J. Monterio, and J. Marquessilva. Assignment and Reordering of Incompletely Specified Pattern Sequences Targeting Minimum Power Dissipation. *12th International Conference on VLSI Design, pp. 37-41, January 1999*

[9] F. Corno, P. Prinetto, M. Rebaudengo, and M. S. Reorda. A Test Pattern Generation Methodology for Low Power Consumption. *16th IEEE VLSI Test Symposium, pp. 453-457, April 1998*

[10] S. Ghosh, S. Basu, and N. A. Touba. Joint Minimization of Power and Area in Scan Testing by Scan Cell Reordering. *Proc. IEEE Annual Symposium on VLSI, pp. 246-249, February 2003*

[11] Latypov P.K. Energy Saving Testing of Circuits. *Automation and Remote Control, April 2001, vol.62, no.4, pp. 653-655(3)*

[12] W. D. Hillis. Co-evolving Parasites Improve Simulated Evolutions as an Optimization Procedure. *Artificial Life II, SFI Studies in the Sciences of Complexity, vol. X., pp. 313-322, 1990*

[13] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer. On average power dissipation and random pattern testability of CMOS combinational logic networks. *Proc. ACM/IEEE Int. Conf. Computer Aided Design, 1992, pp. 402-407*

[14] A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of average switching activity in combinational and sequential circuits. *Proc. ACM/IEEE Int. Conf. Computer Aided Design, 1992, pp. 253-259*

[15] K. Roy and S. Prasad. SYCLOP: Synthesis of CMOS logic for low power application. *Proc. Int. Conf. Computer Design.Oct. 1993, pp. 464-467*

[16] C.-Y. Tsui, M. Pedram and A. Despain. Technology decomposition and mapping targeting low power dissipation. *Proc. ACM/IEEE 30th Design Automation conference, 1993, pp. 68-73*

[17] V. Tiwari, P. Ashar, and S. Malik. Technology mapping for low power. *Proc. ACM/IEEE 30th Design Automation Conference, 1993, pp. 74-79*

[18] H. Vaishnav and M. Pedram. Pcube: A performance driven placement algorithm for low power designs. *Proc. EURO-DAC, Sept. 1993, pp. 72-77*

[19] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch. Efficient Scan Chain Design for Power Minimization During Scan Testing Under Routing Constraint. *International Test Conference, October 2003, pp. 488-493*

[20] S. Narayanan, and M. Breuer. Reconfigurable scan chains: A novel approach to reduce test application time. *Int. Conf. Computer Aided Design, 1993, pp. 710-715*

[21] S. Wang, and S. K. Gupta. DS-LFSR: A new BIST TPG for low heat dissipation. *International Test Conference, October 1997, pp. 848-857*

[22] P. M. Rosinger, P. T. Gonciari, B. M. Al-Hashimi, and N. Nicolici. Simultaneous reduction in volume of test data and power dissipation for systems-on-a-chip. *Electron. Lett., Vol. 37, No. 24, 2001, pp. 1434-1436*

[23] A. Chandra, and K. Chakrabarty. A unified approach to reduce SOC test data volume, scan power, and testing time. *IEEE Trans. Computer-Aided Design, Vol. 20, March 2001, pp. 355-368*

[24] S. Wang, and S. K. Gupta. ATPG for heat dissipation minimization during test application. *IEEE Trans. Comput., Vol. 47, February 1998, pp. 256-262*

[25] R. Sankaralingam, R. R. Oruganti, and N. A. Touba. Static compaction techniques to control scan vector power dissipation. *VLSI Test Symposium, 2000, pp. 35-40*

[26] R. Sankaralingam, and N. A. Touba. Controlling peak power during scan testing. *VLSI Test Symposium, 2002, pp. 153-159*

[27] S. Wang, and S. K. Gupta. An Automatic Test Pattern Generator for Minimizing Switching Activity During Scan Testing Activity. *IEEE Trans. Computer-Aided Design, Vol. 21, No. 8, Aug. 2002, pp. 954-968*

[28] S. Gerstendorfer, and H.-J. Wunderlich. Minimized power consumption for scan-based BIST. *International Test Conference, 1999, pp. 77-84*

[29] T.-C. Huang, and K.-J. Lee. REduction of power consumption in scan-based circuits during test application by an input control technique. *IEEE Trans. Computer-Aided Design, Vol. 20, No. 7, Jul. 2001, pp. 911-917*

[30] N. Parimi, and X. Sun. Toggle-masking for test-per-scan VLSI circuits. *Int. Sym. Defect and Fault Tolerance in VLSI Systems, Oct. 2004, pp. 332-338*

[31] S. Bhunia, H. Mahmoodi, D. Ghosh, S. Mukhopadhyay, and K. Roy. Low-power scan design using first-level supply gating. *IEEE Trans. VLSI, Vol. 13, No. 3. Mar. 2005, pp. 384-395*

[32] L. Whetsel. Adapting scan architectures for low power operation. *International Test Conference, 2000, pp. 863-872*

[33] J. Saxena, K. M. Butler, and L. Whetsel. An analysis of power reduction techniques in scan testing. *International Test Conference, 2001, pp. 670-677*

[34] N. Nicolici, and B. M. Al-Hashimi. Multiple scan chains for power minimization during test application in sequential circuits. *IEEE Trans. Comput., Vol. 51, June 2002, pp. 721-734*

[35] P. Rosinger, B. M. Al-Hashimi, and N. Nicolici. Scan architecture with mutually exclusive scan segment activation for shift- and capture-power reduction. *IEEE Trans. Computer-Aided Design, Vol. 23, No. 7, July 2004, pp. 1142-1153*

[36] H. Weste and K. Eshraghian. Principles of CMOS VLSI Design: A Systems Perspective, 2nd ed. *Reading, MA: Addison-Wesley, 1992*

[37] K. Roy, S. Mukhopadhyay, and H. Mahmoodi. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proc. IEEE, Vol. 91, No. 2, Feb. 2003, pp. 305-327*

[38] D. S. Johnson, C. Aragon, L. McGeoch, and C.Schevon. Optimization by simulated annealing: An experimental evaluation; Part I, graph partitioning. *Operaional Research, Vol. 37, pp. 865-892*

[39] D. Whitley. A Genetic Algorithm Tutorial, *Statistics and Computing (4):65-85, 1994*

[40] T. Blickle and L. Thiele. A Comparison of Selection Schemes used in Genetic Algorithms. *Tech Report No. 11. Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, Zurich, Switzerland, 1995*

[41] B. A. Julstrom and G. R. Raidl. Weight-Biased Edge-Crossover in Evolutionary Algorithms for Two Graph Problems. *Proceedings of the 2001 ACM symposium on Applied computing, 2001, pp.321-326*

[42] D. S. Ha, ATALANTA: An ATPG Tool, Bradley Department of Electrical Engineering, Virginia Polytechnic and State University, Blacksburg, VA, 1994.