Reduction of Power Dissipation during Scan Testing by Test Vector Ordering

Wang-Dauh Tseng[1] and Lung-Jen Lee

Department of Computer Science & Engineering
Yuan Ze University
Chung-li, Taiwan, 32026, ROC
[1]wdtseng@saturn.yzu.edu.tw

Abstract- **Test vector ordering is recognized as a simple and non-intrusive approach to assist test power reduction. Simulation based test vector ordering approach to minimize circuit transitions requires exhaustive simulation of each test vector pair. However, long simulation time makes this approach impractical for circuits with large test set. In this paper we present a calculation based approach to faster order test vectors to reduce test power for full scan sequential circuits. Most calculation approaches are for combinational circuits or for sequential circuits but only considering the portion of circuit derived from the primary inputs. The proposed approach exploits the dependencies between internal circuits and transitions at both the primary and state inputs. Experiments performed on the ISCAS 89 benchmark circuits show that the improvement efficiency of the proposed approach can achieve 91.55% and has better performance than the existing calculation based approaches.**

I. Introduction

Low power electronics has become increasingly important with the advent of portable electronic devices. This has motivated designers to reduce power consumption in the circuit, during both normal operation and testing. Power consumption of digital systems is considerably higher in test mode than in normal mode. During system normal operation, low power consumption can be attributed to the significant correlation that exists between successive vectors applied to a given circuit, whereas in test mode this is not necessarily true. Elevated test power may cause logical error in a fault-free chip leading to an unnecessary loss of yield. Hence it is important to reduce power consumption during test application. Various techniques have been proposed to reduce power consumption during test application [1-7]. Because power consumption in CMOS circuits is proportional to the switching activity in the circuit, the majority of these techniques concentrate on reducing the power consumption by minimizing the switching activity. The technique for power minimization with no penalty in test area and performance is based on test vector ordering which modifies the order in which test vectors of a given test sequence are applied to the CUT [1-5]. The authors in [1] construct a complete weighted graph, called transition graph, in which each vertex represents a test vector and a weight assigned to each edge represents the number of transitions activated in the circuit due to the application of the test vector pair connecting to the edge. Logic and timing simulations are required to compute the weight assigned to each edge. There are totally *n(n–1)/2* simulations required for the construction of the graph, where n is the number of test vectors. The general delay model is assumed during simulations. A greedy algorithm is then used to find a Hamiltonian path of minimum cost in the transition graph. The main problem in this approach rests in the time needed to construct the transition graph for circuits with a large number of test vectors. In order to reduce the graph construction time, the paper in [2] uses zero-delay model for logic and timing simulations. Although the simulation time is reduced, the number of simulations remains *n(n–1)/2* and the time required to construct the transition graph is still high. The paper in [3] propose a fast simulation method which only take into account the expected switching activity at the primary inputs and at a very small set of internal lines of the CUT. The computational time for the construction of transition graph is reduced but the power reduction obtained in [3] is lower than that in [1] and [2]. To make it possible to apply test vector reordering to circuits with large number of test vectors, the authors in [4] employ the Hamming distance between test vectors rather than simulate transitions in the circuit to evaluate the power consumption. Although the Hamming distance approach is the most time-saving, it doesn't take into account the topology of the circuit. The reduction in power consumption for the Hamming distance approach is usually significantly lower than that for the simulation based approaches [1-3]. The authors in [5] consider the structure of the CUT in the estimation of the weights in the transition graph thus providing better solutions in terms of amount of power saved. An induced activity function is proposed in this paper to measure the impact of a transition at a specified input on the switching activity of the CUT and is used as the weight of the Hamming distance. However, these calculation based approaches in [2-5] are only applicable to combinational circuits or full scan sequential circuits with specific scan cell design. In this paper we present a calculation based test vector ordering technique that reduces power consumption during test application for full scan sequential circuits. We exploit the dependencies between internal circuits and the transitions at scan cells as well as primary inputs. The idea behind the proposed approach is based on the following two observations. First, for a scan cell, the number of transitions caused by a test vector being scanned in/out depends not only on the transitions in the test vector/response but also on its position in the scan chain. Second, the impact of each circuit input on the switching activity in the internal circuit is different. If a transition at a circuit input of the CUT propagates to the internal circuit, it

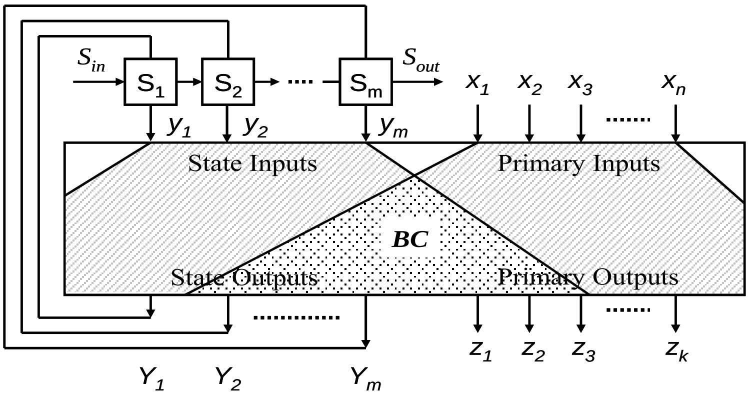

Fig. 1 A full scan sequential circuit.

will result in a large number of unnecessary transitions. Depending on the circuit structure, the transitions at some circuit inputs may cause more transitions at internal circuit than those at other circuit inputs. Based on the two observations, weights on the transition graph for test vector ordering can be derived. In the following, two functions to compute the transition count of a scan cell in the scan chain and to measure the impact of a circuit input on switching activity in the internal circuit are developed, respectively. The rest of the paper is organized as follows. Section 2 provides the background on scan testing and some circuit definitions. In Section 3, an impact function is developed to measure the impact of transitions at a gate on the switching activity in the combinational part of the circuit. Section 4 calculates the transition count at each scan cell as a pair of test vectors is scanned. Section 5 presents the proposed approach. To validate the proposed approach, experimental results are given in Section 6. Section 7 concludes the paper.

II. Circuit definitions

Consider the full scan sequential circuit shown in Fig.1, comprised of a block of combinational circuit C and a set of m state elements. The primary inputs and outputs of the circuit are x_1, x_2,…, x_n, and z_1, z_2,…, z_k, respectively. The present state variables, y_1, y_2,…, y_m, constitute the state inputs of the combinational circuit. The next state variables, Y_1, Y_2,…, Y_m, constitute the state outputs of the combinational circuit. The *forward cone* of line *l*, *FC*(*l*), is defined as the portion of a circuit whose signals are reachable by a forward trace of the circuit topology starting at *l*. The *primary forward cone* (*PFC*) is defined as the portion of the circuit which is the union of forward cones of all primary inputs. The change of values on primary inputs by applying consecutive test vectors causes switching activity in the primary forward cone. The *state forward cone* (*SFC*) is defined as the portion of a circuit which is the union of forward cones of all state inputs. The change of values on state inputs by shifting a scan vector causes switching activity in the state forward cone. The portion of the circuit defined as the intersection of primary forward cone and state forward cone is referred to as *blocking cone* (*BC*). Any gate in the blocking cone has at least one input in the path which starts from a primary input and one input in the path which starts from a state input. Most calculation based test vector ordering approaches to reducing test power is by lowering the transition density at the primary inputs. Very little work considers reducing the transition density at the state inputs; therefore, only the switching activity in the PFC of the CUT can be reduced by these approaches. In this paper, we try to lower the transition density at both the primary inputs and state inputs by ordering the test vectors such that the switching activity in the PFC and SFC of the CUT can be reduced.

III. Calculation of Impact Function

During scan cycle, filling in the scan chain with the state input part of a test vector requires shifting the bits one by one into each scan cell, thus creating increased switching activity in the scan cells. The rippling effect originating from a scan chain to the CUT results in a large number of unnecessary transitions at the circuit lines. Additionally, if a transition at a primary input of the CUT propagates to the internal circuit, it will subsequently cause transitions. Depending on the circuit structure, the transitions at some circuit inputs (primary /state inputs) of a CUT cause more transitions at internal lines than those at other circuit inputs. Therefore, reducing transitions at those circuit inputs that cause more transitions in the internal circuit will make greater reduction in switching activity. To measure the transition effect at a circuit input reflecting into the CUT, it is necessary to develop an approach to evaluate the impact on the transitive fanout of each circuit input. The authors in [7] propose a gain function for computing the weighted transition density, which provides an effective measure of the switching activity in logic circuit, for each primary input. In this section, we develop an impact function, which is modified from the gain function, to measure the transition of a state/primary input impact on the switching activity in the internal circuit.

Consider a CUT with *m* circuit inputs x_1, x_2, …, x_m. The *signal probability* *sp*(*c*) of a circuit line *c* is defined as the probability that *c* is set to 1:

$$sp(c) = Pr(c = 1) \qquad (1)$$

Signal probability can be propagated through logic gates based on simple rules of probability and logic function of the gates. The *transition probability* of a circuit line *c* is the probability of the signal making a transition from one state to another at any time *t* and is denoted by $p_t(c)$. Under the assumption that the values applied to each circuit input are temporally independent, we can write:

$$p_t(c) = 2 \cdot sp(c) \cdot (1 - sp(c)) \tag{2}$$

Let $n_c(T)$ be the number of transitions at a circuit line c in a time interval of length T. The *transition density* at c, i.e. the number of transitions per second at c, is defined as:

$$D(c) = \lim_{T\to\infty} \frac{n_c(T)}{T} \tag{3}$$

Let f_i be a function that depends on circuit input x_i. The Boolean difference of f_i with respect to x_i is defined as follows:

$$\frac{\partial f_i}{\partial x_i} = f_i\,|_{x_i=1} \oplus f_i\,|_{x_i=0} \tag{4}$$

where $\oplus$ denotes the exclusive-or operation. The Boolean difference signifies the condition under which output f_i is sensitized to circuit input x_i. If the circuit inputs x_i, $i = 1, \ldots, m$, to the CUT are not spatially correlated, the transition density of a circuit line c can be defined in terms of the Boolean difference with respect to each circuit input, $\partial f_i / \partial x_i$, and the transition density of each circuit input, $D(x_i)$, as:

$$D(c) = \sum_{i=1}^{m} P\left(\frac{\partial f_i}{\partial x_i}\right) D(x_i) \tag{5}$$

The Boolean difference $\partial f_i / \partial x_i$ represents the condition for sensitizing circuit input x_i to output f_i as noted in Eq. (4). Therefore, $P(\partial f_i / \partial x_i)$ signifies the probability of sensitizing input x_i to output f_i, while $P(\partial f_i / \partial x_i)D(x_i)$ is the contribution of transitions at output f_i due to circuit input x_i only. Hence, the contribution of transitions at output f_i due to all the circuit inputs is obtained by taking the summation over all the circuit inputs of the CUT.

As shown in Eq. (5), the transition density of a circuit line c is the sum of the transitions at each circuit input that sensitize to line c. Hence, the portion of the transition density of line c due to the transition at a circuit input x_i, is given by

$$D_{x_i}(c) = P\left(\frac{\partial f_i}{\partial x_i}\right) D(x_i) \tag{6}$$

Similarly, the portion of the transition density of line c due to the transition at a specific line k, can be written by

$$D_k(c) = P\left(\frac{\partial g_i}{\partial k}\right) D(k) \tag{7}$$

where g_i is a function that depends on circuit line k. The sum of transition densities of lines in the forward cone of k, $FC(k)$, that can be attributed to the transitions at k is given by:

$$D_k = \sum_{\forall c \in FC(k)} D_k(c) \tag{8}$$

For the CMOS circuit technology, dynamic power due to the charging and discharging circuit capacitances is the dominant source of power consumption. Hence, the power dissipation in a circuit depends on the load capacitance of internal lines. However, lines connected to more gates are lines with higher parasitic capacitance. If two circuit lines have the same transition density, the one with higher fanout will consume more power than the other one with lower fanout. Load capacitance also depends on the type and the size of the device. For example, a 2-input NOR has more load capacitance than a 2-input NAND, and the load capacitance difference increases as the number of inputs increase. So, two factors, the fanout and device coefficient, are also considered in the impact function as the weights of the transition density. Although wire length also affects power consumption, for simplicity, it is not considered in the impact function. For each circuit line k the impact function IMP_k can be expressed as:

$$IMP_k = \sum_{\forall c \in FC(k)} F_c \alpha_c D_k(c) \tag{9}$$

where F_c and α_c are the fanout and device coefficient of circuit line c, respectively. The fanout of the lines is defined by circuit topology. The device coefficient can be obtained once the circuit has been synthesized. In this derivation, the Boolean difference of f_i with respect to c is derived from the signal probability. However the signal probability $P(\partial f_i / \partial sp_i)$ can easily be calculated using the similar procedure which is used to calculate detection probability [3].

To illustrate the calculation of impact function value, considers the circuit shown in Fig. 2. The primary inputs are $\{x_1, x_2, x_3, x_4\}$, $\{S_1, S_2, S_3, S_4\}$ are scan cells, $\{y_1, y_2, y_3, y_4\}$ are the state inputs, and $\{z_1, z_2\}$ are the primary outputs. The forward cone of primary input x_3 is composed of circuit lines c_1, c_2, c_5, c_7, c_8 and c_{10}. Table 1 shows the impact function value for primary input x_3 in Fig. 2. The first row shows the circuit lines in the forward cone of x_3. The second and third rows show the corresponding fanout and transition density of each circuit line in each column, respectively. The circuit lines and their fanouts included in the forward cone of x_3 can be obtained directly from the circuit structure. However the transition density for each circuit line can be calculated using Eqs. (4) and (6). Take circuit line c_5 for example. As shown in Fig. 2, the Boolean function f_{c5} for circuit line c_5 can be express as $\overline{(x_2x_3 + y_2)x_2}$. By the definition in Eq. (4) the Boolean difference of f_{c5} with respect to x_3 can be expressed as $\frac{\partial f_{c5}}{\partial x_3} = \overline{(x_2x_3 + y_2)x_2}\big|_{x_3=0} \oplus \overline{(x_2x_3 + y_2)x_2}\big|_{x_3=1} = x_2\overline{y_2}$. Hence the signal probability for c_5, $f\left(\frac{\partial f_{c5}}{\partial x_3} = x_2\overline{y_2}\right)$, is equal to 1/4. Assume that the signal probability for each bit in a test vector is 1/2. By Eq. (2) the transition probability for x_3 is equal to 2 × 1/2 × (1 - 1/2) = 1/2. According to Eq. (6), the transition density for c_5 is equal to 1/4 × 1/2 = 1/8. Assuming that the device coefficient is 1 for all type of gates, the impact function value of x_3 can be calculated by Eq. (9) and is equal to 1 × 1/2 + 2 × 1/4 + 1 × 1/8 + 1 × 1/16 + 2 × 1/16 + 1 × 3/32 = 45/32 = 1.4.

Table 1 Calculation of impact function value for primary input x_3.

FC(x_3)	c_1	c_2	c_5	c_7	c_8	c_{10}	*IMP*
Fanout	1	2	1	1	2	1	1.4
TD	1/4	1/4	1/8	1/16	1/16	3/32	

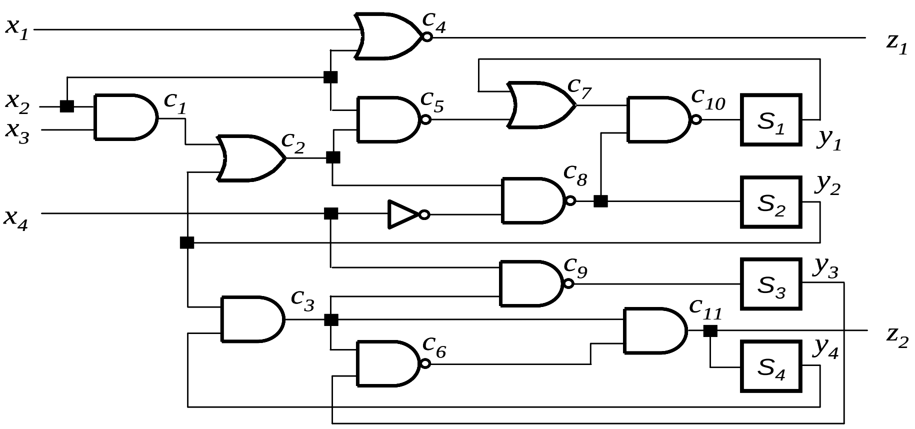

Fig. 2 An example circuit.

IV. Scan Cell Transition Count Calculation

The paper in [6] has developed an expression, called weighted transition count (WTC), to compute in the scan chain the number of transitions caused by a test vector being scanned in. The expression is based on the observation that the number of scan cell transitions caused by a transition in a test vector being scanned in depends on its position in the test vector. The number of weighted transitions is given by:

Weighted_Transitions = ∑ (Size_of_Scan_Chain – Position_of_Transition) (10)

It is shown that the weighted transition count is very well correlated with the real power dissipation and hence the power dissipated when applying two vectors can be compared by counting the number of weighted transitions in the vector. However, the WTC with respect to a test vector can also be computed by counting the transitions at each scan cell. For example, consider a scan chain with five scan cells (d_1, d_2, d_3, d_4, d_5) and the test vector $b_5b_4b_3b_2b_1$ = 00101 being scanned in. The test vector has 3 transitions, T_1 between b_1 and b_2, T_2 between b_2 and b_3, and T_3 between b_3 and b_4. By Eq. (1), WTC can be computed by summing up the weight of each transition in the test vector; that is, WTC = (5-1) + (5-2) + (5-3) = 9. The same result can also be obtained by summing up the number of transitions at each scan cell in the scan chain. In the following, we compute the number of transitions at each scan cell. Note that not all the bits in a test vector will pass through every scan cell. For instance, only subsequences b_2b_1 and $b_4b_3b_2b_1$ will pass through the scan cells d_4 and d_2, respectively. To put it more formally, given a scan chain with m scan cells, the subsequence of a test vector to pass through the scan cell d_i can be expressed as $b_{m-i+1}b_{m-i}\ldots b_1$ where $1 \leq i \leq m$. Therefore, the number of transitions at a scan cell can thus be calculated by counting the number of transitions in the subsequence that passes through it. Continuing above example, for the first scan cell d_1 there are 3 transitions in $b_5b_4b_3b_2b_1$; hence the transition count for scan cell d_1 is 3. Similarly, there are 3, 2, 1, and 0 transitions in the subsequences $b_4b_3b_2b_1$, $b_3b_2b_1$, b_2b_1, and b_1, respectively. Hence the transition counts for scan cells d_2, d_3, d_4, and d_5 are 3, 2, 1, and 0, respectively. The WTC can thus be calculated by summing up the transition count of each scan cell in the scan chain, i.e. 3 + 3 + 2 + 1 + 0 = 9. This value is the same as the value obtained by Eq. (1). What we have explained above reveals that the number of transition count at a scan cell depends not only on the number of transitions in the test vector but also on its position in the scan chain. The number of transition count caused by scanning in test vector V_t at the scan cell d_i is given by:

$$TC_{scanin}(d_i) = \sum_{j=1}^{m-i} \left(V_t(j) \oplus V_t(j+1)\right) \qquad (11)$$

where m is the length of the scan chain and $V_t(j)$ represents the j^{th} bit of the test vector V_t. Similar reasoning can be applied to scan out test responses. The number of transition count caused by scanning out test response V_r at the scan cell d_i is given by:

$$TC_{scanout}(d_i) = \sum_{j=1}^{i-1} \left(V_r(j) \oplus V_r(j+1)\right) \qquad (12)$$

The WTC value corresponding to V_k scan-in or scan-out can be obtained by summing up transition counts of each scan cell in the scan chain and is given by:

$$WTC(V_k) = \sum_{i=1}^{m} TC(d_i) \qquad (13)$$

V. Proposed Approach

Test vector ordering is equivalent to a permutation of a given set of test vectors. All the test vector ordering techniques in [1-5] are based on the construction of a transition graph with a weight associated with each edge. The weight represents the power consumed after consecutive test vectors are applied. The problem is then amounted to finding a Hamiltonian path of minimum cost in the transition graph. The main difference between these approaches is the representation of the weight. In the following we present a method to calculate the weight for each pair of test vectors. Since the power consumed during scan testing can be divided into two major parts: power during shift cycle and power during application cycle, the calculation of the weight can also be divided in two parts: weight for state input part and weight for primary input part. First, we consider the calculation of weight for the state input part. Suppose that each test vector for the CUT has n + m bits where the first n bits form the primary input part and the last m bits form the state input part. When applying a test vector pair (V_{k-1}, V_k) to the CUT, the state output of V_{k-1} is scanned-out simultaneously with the scanning-in of the state input of test vector V_k. And the transition count of a scan cell d_i , $TC(d_i)$, in the scan chain after applying the test vector pair (V_{k-1}, V_k) can be expressed as:

$$TC(d_i) = \sum_{j=m+1}^{m+n-i} \left(R_{k-1}(j) \oplus R_{k-1}(j+1)\right) + \sum_{j=m+1}^{i-m-1} \left(V_k(j) \oplus V_k(j+1)\right) \tag{14}$$

where $R_{k-1}(j)$ represents the j^{th} bit of the state output after applying test vector V_{k-1} and $V_k(j)$ represents the j^{th} bit of the state input of test vector V_k. As described previously, an impact function is used to measure the impact of a circuit input on the switching activity in the internal circuit. The circuit input with higher impact function value has higher probability to cause more transitions in the CUT. Hence, the weight which takes into account dependencies between internal nodes and the transitions on the state input sp_i can thus be defined as $TC(d_i) \times Isp_i$. Therefore, the weight for the state input part on each edge (V_{k-1}, V_k) representing the cost in terms of transition density of the application of the test vector pair (V_{k-1}, V_k) to the state forward cone can be defined as:

$$Weight\_sip(V_{k-1}, V_k) = \sum_{i=m+1}^{n+m} TC(d_i) \times Isp_i \tag{15}$$

where Isp_i is the impact function value of state input sp_i and $TC(d_i)$ is the transition count of scan cell d_i after applying the test vector pair (V_{k-1}, V_k).

Secondly, we consider the calculation of weight for the primary input part. If a transition at a primary input of the CUT propagates to the internal circuit, it will subsequently cause more transitions. When applying the state input parts of two consecutive test vectors, say (V_{k-1}, V_k), to the CUT, the transition occurred at a primary input p_i, $TC(p_i)$, can be represented as:

$$TC(p_i) = \left(V_{k-1}(i) \oplus V_k(i)\right) \tag{16}$$

Hence, the weight which takes into account dependencies between internal nodes and the transitions on the primary input p_i can thus be defined as $TC(p_i) \times Ip_i$. The weight for the primary input part on each edge (V_{k-1}, V_k) representing the cost in terms of transition density of the application of the test vector pair (V_{k-1}, V_k) to the primary forward cone can be defined as:

$$Weight\_pip(V_{k-1}, V_k) = \sum_{i=1}^{n} TC(p_i) \times Isp_i \tag{17}$$

The total weight on edge (V_{k-1}, V_k) can be formally defined as:

$$\begin{aligned} Weight(V_{k-1}, V_k) &= Weight\_pip(V_{k-1}, V_k) + Weight\_sip(V_{k-1}, V_k) \\ &= \sum_{i=1}^{n} TC(p_i) \times Isp_i + \sum_{j=m+1}^{n+m} TC(d_j) \times Isp_j \end{aligned} \tag{18}$$

Similar to the other test vector ordering approaches in [1-5], construction of a transition graph is required in this paper. Consider a set of test vectors $S = (V_1, V_2, \ldots, V_n)$. We construct a complete directed graph $G = (\varphi, E)$, where each vertex $V_k \in \varphi$ represents a test vector of S and each directed edge $(V_i, V_j) \in E$ represents a pair of test vectors. The weight on each edge (V_i, V_j) of G is calculated using Eq. (18). Then the test vector ordering problem is amounted to finding a Hamiltonian path of minimum cost in the complete graph, which is known to be an NP-hard problem. A rich literature exists on the algorithms for solving this problem. We develop a program based on the algorithm in [8] to determine the order of the test vectors that minimizes the power consumed during test.

VI. Experimental Results

To validate the proposed approaches, we have carried out experiments on full scan versions of the ISCAS 89 benchmark circuits. Table 2 shows basic characteristics of these circuits. They are gate counts, number of Flip-Flops, primary inputs and primary outputs, respectively. The procedure to determine the order of the test vectors was implemented on a 1.5 GHz Pentium IV PC with 512 MB RAM running Linux and using GNU CC version 2.19. We use average transition count (ATC) as quantitative measure for power consumption. The ATC is defined as the total number of transition counts divided by the total number of clock cycles. To evaluate the effectiveness of the proposed vector ordering approach, we compare the ATC and the improvement of transition reduction with that of the simulation based approach. Simulation based approach is founded on exhaustive counting of the transitions at all circuit lines for every test pair. The calculation based test vector ordering approach in paper [5] are also implemented and are compared with the proposed approach. The first column gives the circuit name. In the second column, the first sub-column *Original* shows the ATC for the circuits without using any vector ordering techniques. The second, third, and fourth sub-columns (*Our*, *Paper*[5], and *Simu.*) show the ATC by using the proposed approach, the approach in paper [5], and the simulation based approach, respectively. The next column shows the improved percentages of ATC for the proposed

Table 2 Characteristics of ISCAS 89 benchmark circuits

Circuit	Gates	FF's	PI's	PO's
s298	119	14	3	6
s344	160	15	9	11
s420	218	16	19	2
s510	211	6	19	7
s641	379	19	35	24
s713	393	19	35	23
s1423	657	74	17	5
s5378	2779	179	35	49
s9234	5597	211	36	39
s13207	7951	638	62	152
s15850	9772	534	77	150
s35932	16065	1728	35	320
s38417	22179	1426	38	304

Table 3. Results of power reduction

Circuit	Average Transition Count				Improvement (%)			Efficiency (%)		CPU Time (Sec.)
	Original (A)	*Ours* (B)	*Paper*[5] (C)	*Simu.* (D)	*Ours* $\frac{(A)-(B)}{(A)}$ ×100	*Paper*[5] $\frac{(A)-(C)}{(A)}$ ×100	*Simu.* $\frac{(A)-(D)}{(A)}$ ×100	*Ours*	*Paper* [5]	
s298	140.61		132.732	130.06	7.37	5.60	7.5	98.27	74.70	0.14
s344	293.38		273.207	265.10	8.84	6.88	9.64	91.70	71.33	0.31
s420	307.3		283.681	270.49	9.86	7.69	11.98	82.30	64.16	0.83
s510	314.42		269.239	258.61	16.6	14.37	17.75	93.52	80.96	0.34
s641	426.65		383.615	374.39	10.73	10.09	12.25	87.59	82.34	1.17
s713	409.48	362.27	366.042	358.01	11.53	10.61	12.57	91.73	84.39	1.21
s1423	1019.25	937.00	953.875	933.84	8.07	6.41	8.38	96.30	76.54	10.75
s5378	3883.39	3580.49	3656.18	3557.96	7.8	5.85	8.38	93.08	69.82	38.54
s9234	6758.39	6335.99	6468.31	6315.04	6.25	4.29	6.56	95.27	65.43	76.42
s13207	12579.98	130.25	12109.3	11805.05	5.59	3.74	6.16	90.75	60.74	109.24
s15850	11079.91	267.45	10681.6	10427.30	5.21	3.59	5.89	88.46	61.03	98.87
s35932	22784.8	277.00	22142.2	21613.66	4.83	2.82	5.14	93.97	54.87	35.29
s38417	33023.1	262.23	32308.7	31652.64	3.62	2.16	4.15	87.23	52.13	1473.2
Avg.	7155.44	380.87	6925.28	6766.32	8.18	6.47	8.95	91.55	69.11	142.03

approach, the approach in paper [5], and the simulation based approach, respectively. Compared with the approach without using any vector reordering techniques, only 8.18% of power reduction is improved for the proposed approach. This is because lots of the circuits in Table 3 are controlled mostly by scan chains rather than primary inputs. However, the test vector ordering approach is independent to a lot of approaches such as circuit modification, scan chain partitioning, and scan cell ordering, so it can be utilized together with these approaches to further reduce test power. The last column compares the proposed approach with the simulated approach in terms of *improvement efficiency* which is defined as the ratio between the improved percentages obtained by the proposed approach (or the approach in [5]) and by the simulation based approach. From Table 3, we find that the percentage improvement achieved by the proposed approach is close to the simulation based approach and the improvement efficiency can achieve 91.55% on average and always has better results than the approach in [5] in all cases. In simulation based approach the calculation of ATC for every possible pair of test vectors requires logic simulation. When the test set is large the required computation time is prohibitively long, although it has better power reduction rate than that of the calculation based approach.

VII. CONCLUSIONS

We have presented a calculation based test vector ordering approach to reduce power dissipation for full scan sequential circuits. The proposed approach exploits the dependencies between internal circuits and the transitions at circuit inputs. Two functions are developed to compute the transition weight of a scan cell and to measure the impact of transitions at a circuit input on switching activity in the internal circuit, respectively. Experimental results for benchmark circuits show that improvement efficiency can achieve 91.55% on average.

REFERENCES

[1] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, Techniques for minimizing power dissipation in scan and combinational circuits during test application, *IEEE Trans. Computer-Aided Design*, Dec.1998, vol. 17, pp. 1325-1333,.

[2] Z. Luo, X. Li, H. Li, S. Yang, and Y. Min, Test Power Optimization Techniques for CMOS Circuits, 11th *Asia Symposium*, 2002, pp. 332-337.

[3] X. Kavousianos, D. Bakalis, M. Bellos, and D. Nikolos, An Efficient Test Vector Ordering Method for Low Power Testing, *Proceedings IEEE Computer Society Annual Symposium on VLSI*, Feb. 2004, pp. 285-288.

[4] P. Girard, C. Landrault, S. Pravossoudovitch, and D. Severac, Reducing power consumption during test application by test vector ordering, *Proc. IEEE int'l Symp. Circuits and Systems*, part II, 1998, pp. 296-299.

[5] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, A Test Vector Ordering Technique for Switching Activity Reduction during Test Operation, *Proceedings Ninth Great Lakes Symposium on VLSI*, Mar. 1999, pp. 24-27.

[6] 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.

[7] S. Wang and S. K. Gupta, DS-LFSR: a BIST TPG for low switching activity, *IEEE Trans. Computer-Aided Design*, July 2002, vol. 21, pp. 842-851.

[8] M. N. Swamy and K. Thulasiraman, *Graphs, Networks, and Algorithms*. New York: Wiley, 1981.