Low Power Testing by Test Vector Ordering with Vector Repetition†

M. Bellos[1,2], D. Bakalis[1,2], D. Nikolos[1,2] and X. Kavousianos[3]
[1]Computer Engineering & Informatics Department, University of Patras, 26500 Patras, Greece
[2]Research Academic Computer Technology Institute, 61 Riga Feraiou St., 26221 Patras, Greece
[3]Computer Science Department, University of Ioannina, 45110 Ioannina, Greece
bellos@ceid.upatras.gr, {bakalis, nikolosd}@cti.gr, kabousia@cs.uoi.gr

Abstract

Test vector ordering with vector repetition has been presented as a method to reduce the average as well as the peak power dissipation of a circuit during testing. Based on this method, in this paper we present some techniques that can be used to further reduce the average power dissipation. Experimental results validate that the proposed techniques achieve considerable savings in energy and average power dissipation while reducing the length of the resulting test sequences compared to the original method.

1. Introduction

During the last decade, several power optimization techniques targeting minimal switching activity at circuit nodes have been proposed at all levels of the design hierarchy so as to reduce power dissipation. On the other hand, the test efficiency has been shown to have a high correlation with the toggle rate; hence, in test mode, in order to excite many potential faults by relatively short test sequences, the switching activity at circuit nodes is often several times higher than the switching activity during normal operation [1]. The elevated average and peak power dissipation during testing can be responsible for several kinds of problems: decreased overall yield, decreased reliability and system life cycle and increased product costs [1-2]. In battery powered systems, the energy dissipated during BIST-based periodic testing is also a significant issue.

Several techniques have been proposed during the last few years for minimizing switching activity during test [1-6]. Among them, post-ATPG test vector ordering techniques have been presented in [3-6]. The basic idea beyond test vector ordering is to reduce the switching activity of the circuit under test (CUT) by reducing the switching activity at the inputs of the circuit during testing. Test vector ordering was proven to be NP-hard and equivalent to the Traveling Salesman Problem, hence heuristics are used to solve the problem [3].

† This research was financially supported by the Public Benefit Foundation "Alexander S. Onassis" via its scholarships programs and by the State Scholarships Foundation of Greece by its Post-Doctoral research scholarships program.

Test vector ordering is equivalent to a permutation of a given set of test vectors. It has been shown that if the restriction of using each test vector only once is relaxed, then the average power [3] as well as the peak power [7] dissipation can be further reduced. The authors of [3,7] have given a heuristic, which reduces the average and peak power dissipation during testing while guaranteeing that the length of the final sequence is no more than twice the length of the original sequence. However, the length of the resulting sequence, hence the energy dissipation during testing, depends on some parameters which have not been investigated in [3].

In this paper we present two techniques that can be used along with the method presented in [3]. The first technique modifies some of the vectors of the test sequence derived from the application of [3] so as to reduce the average power as well as the energy dissipation during testing whereas the second technique aims to reduce the length of the test sequence, hence the energy dissipation. We then evaluate the effect of a parameter that has not been investigated in [3]. Finally, we compare the proposed techniques with the method presented in [3] and confirm using experimental results that significant power and energy savings can be achieved while the length of the resulting test sequences is reduced.

2. Test Vector Ordering with Vector Repetition

In this paper we assume the zero-delay model. It has been shown in [8] that there is a correlation between the energy dissipation of a circuit assuming a zero delay and the energy dissipation assuming a general delay model. Hence, using a zero delay approximation is reasonable. Consider a combinational circuit C and a set of test vectors $TV = \{tv_1, \ldots, tv_n\}$ with cardinality n. We construct a complete undirected graph $TG = (V, E)$ where each vertex $v_i \in V$ corresponds to a test vector $tv_i \in TV$. Each undirected edge $(v_i, v_j) \in E$ represents a pair of test vectors and is assigned a weight $w(v_i, v_j)$. According to the research so far presented in the open literature, weight $w(v_i, v_j)$ can be equal to:

(a) the number of transitions activated in the CUT due to the application of test vector pair (tv_i, tv_j) [3, 6], or

(b) the Hamming distance of test vector pair (tv_i, tv_j) [4], or

(c) the sum of *induced activity function* values for all primary inputs that change from 1 to 0 or from 0 to 1 after application of test vector pair (tv_i, tv_j) [5]. The *induced activity function* is calculated for every primary input of the CUT and is equal to the sum of the transition densities of all nodes in the circuit that can be attributed to the transitions at that primary input.

If we denote as $P(a, b)$ the number of transitions activated in the circuit after the application of test vector pair (a, b), then the problem of test vector ordering with vector repetition can be stated as follows [3]: *Given a combinational circuit C and a set of test vectors* $TV = \{tv_1, ..., tv_n\}$, *compute an optimal input sequence* $S = <s_1, ..., s_m>$, *where* $m \geq n$ *and* $\forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \exists s_j$ *such that* $s_j = tv_i$ *and* $\frac{1}{m-1}\sum_{i=1}^{m-1} P(s_i, s_{i+1})$ *is minimized*. If we restrict the length of the final sequence to be no more than twice the length of the original sequence, then we have an additional restriction that $m/n \leq 2$.

A heuristic, denoted hereafter as TVO_VR, for this problem has been presented in [3] and will be described briefly in the following. This heuristic is based on the use of the minimum spanning tree (MST) of a graph. Consider the undirected graph *TG* that is constructed from test set *TV*, and that weights have been assigned to edges using one of the above metrics. Kruskal's algorithm is used to construct an MST of the graph. A parameter *threshold* is set equal to the average value of weights of the edges belonging to the MST. Then by choosing an arbitrary vertex as the root we get the inorder traversal of the tree. Next, we produce a tour of the tree's edges as follows: starting from the leftmost vertex of the inorder traversal, designated as *current*, we select the next vertex of the inorder traversal, designated as *next*. If *current* and *next* are connected by an edge of the spanning tree or if their edge has a weight smaller than *threshold* then we add *next* to the Tour and set *current* equal to *next*. Otherwise, we backtrack in the tree by one step by setting *current* equal to the parent of *current* and repeat until the above condition applies. This is actually the step where the vectors are repeated. We then continue with the next vertex of the inorder traversal until all vertices are covered.

3. Proposed techniques

In this section we first present a new technique, which can be used along with TVO_VR to further reduce the average power as well as the energy dissipation during testing. Then, we propose a method for root selection and rearrangement of the MST aiming the reduction of the length of the final test sequence and hence the energy dissipation. Finally, we consider the influence of the value of the parameter *threshold* on energy and average power reduction as well as test sequence length.

v_i	1	0	0	0	1	1	1	0
r_1	0	0	1	1	0	0	1	1
r_2	1	1	0	0	1	0	0	0
r_3	0	1	1	1	1	1	1	1
v_j	1	1	0	1	0	1	0	0

$\Rightarrow$

v_i	1	0	0	0	1	1	1	0
r'_1	1	0	0	1	0	1	1	0
r'_2	1	1	0	0	1	1	0	0
r'_3	1	1	0	1	1	1	1	0
v_j	1	1	0	1	0	1	0	0

Figure 1. An example of modifying repeated vectors

3.1. Modifying repeated vectors

The TVO_VR algorithm [3] produces a sequence of at most $2n$ test vectors. We can divide the vectors of this sequence into two sets: (a) the set of test vectors $FA = \{fa_1, \ldots, fa_n\}$, with $fa_i \in TV$, consisting of the vectors that appear for the first time in the sequence, and (b) the set of test vectors $R = \{r_1, \ldots, r_k\}$, $k \leq n$, consisting of the remaining vectors of the sequence. The vectors belonging to R, henceforth called repeated vectors, were inserted into the sequence by the backtracking step of the algorithm so that the average power dissipation is reduced. Obviously, since: (a) sets FA and TV are equal by construction and (b) the vectors of set R are chosen from the original test set, for every test vector $r_i \in R$ there exists a vector $fa_j \in FA$ such that $r_i = fa_j$. That is, the repeated vectors do not contribute to the final fault coverage since the faults they cover are also covered by the test vectors of set FA. Therefore, we can modify the repeated vectors so as to reduce the number of transitions at the primary inputs of the CUT leading to reduction of the number of transitions at internal lines hence lowering power dissipation during testing.

For example, consider a circuit with 8 primary inputs. Let us assume that between test vectors v_i and v_j (v_i, $v_j \in FA$) the algorithm inserted 3 vectors r_1, r_2, $r_3 \in R$ (see Figure 1). Consider the leftmost bit of every test vector. Vectors v_i and v_j have the same value (1) whereas vectors r_1, r_2, r_3 have values 0, 1, and 0 respectively. Consequently, these 5 test vectors, when applied to the circuit, will produce 4 transitions at the corresponding primary input. However, since vectors r_1, r_2, r_3 belong to set R we can modify them and assign value 1 to the leftmost bit of all 3 vectors. In this case, the number of transitions in this bit will be reduced to 0. If we repeat this procedure on every bit position where vectors v_i and v_j have the same value, we can reduce the total number of transitions at the primary inputs from 24 to 10.

A more formal description of the procedure for modifying the repeated vectors (henceforth called ModifyRepeatedVectors or MRV) follows. The procedure receives as input the sequence of vectors produced by TVO_VR algorithm. Each vector s_i of the sequence has a flag associated with it, which identifies if the vector is member of the FA set or not. The TVO_VR algorithm guarantees by construction that the first (s_1) and last (s_m) vector of the sequence belong to the FA set. Therefore, the ModifyRepeatedVectors procedure begins from vector s_1 and proceeds until it has reached the last vector of the sequence s_m. At each step, the procedure checks for the existence of repeated vectors between two "neighboring"

Table 1. Power savings with MRV procedure (graph constructed as proposed in [4])

Circuit	Test sequence length			TVO_VR			TVO_VR with MRV			Avg. Power & Energy Reduction	Peak Power Reduction
	Original	TVO_VR	Factor	Energy	Avg. Power	Peak Power	Energy	Avg. Power	Peak Power		
c432	52	99	1.90	8405	84.9	138	6249	63.1	137	25.7%	0.7%
c499	59	101	1.71	7067	70.0	146	6513	64.5	141	7.8%	3.4%
c880	49	89	1.82	21266	238.9	341	17730	199.2	341	16.6%	0.0%
c1355	86	162	1.88	29275	180.7	320	27655	170.7	313	5.5%	2.2%
c1908	117	223	1.91	90423	405.5	608	76229	341.8	608	15.7%	0.0%
c2670	92	180	1.96	88298	490.5	878	68258	379.2	878	22.7%	0.0%
c3540	206	404	1.96	281876	697.7	1104	229104	567.1	1104	18.7%	0.0%
c5315	114	225	1.97	375274	1667.9	2112	282774	1256.8	2088	24.6%	1.1%
c6288	27	48	1.78	76154	1586.5	2064	67030	1396.5	2039	12.0%	1.2%
c7552	162	318	1.96	596457	1875.7	3084	464627	1461.1	3084	22.1%	0.0%

Table 2. Power savings with MRV procedure (graph constructed as proposed in [5])

Circuit	Test sequence length			TVO_VR			TVO_VR with MRV			Avg. Power & Energy Reduction	Peak Power Reduction
	Original	TVO_VR	Factor	Energy	Avg. Power	Peak Power	Energy	Avg. Power	Peak Power		
c432	52	99	1.90	8364	84.5	140	6394	64.6	135	23.6%	3.6%
c499	59	110	1.86	7038	64.0	164	6338	57.6	164	9.9%	0.0%
c880	49	86	1.76	21199	246.5	329	17073	198.5	329	19.5%	0.0%
c1355	86	165	1.92	29552	179.1	313	27216	165.0	313	7.9%	0.0%
c1908	117	226	1.93	88033	389.5	596	76203	337.2	596	13.4%	0.0%
c2670	92	175	1.90	83025	469.1	820	64289	363.2	820	22.6%	0.0%
c3540	206	404	1.96	253459	627.4	993	202849	502.1	993	20.0%	0.0%
c5315	114	224	1.96	360194	1608.0	1975	265054	1183.3	1945	26.4%	1.5%
c6288	27	50	1.85	81092	1621.8	2064	68338	1366.8	2064	15.7%	0.0%
c7552	162	305	1.88	537289	1761.6	2531	416421	1365.3	2531	22.5%	0.0%

vectors that belong to set *FA* (denoted as *CurrentVector* and *NextVector*). In case repeated vectors exist, then in every bit position that *CurrentVector* and *NextVector* have the same value (0 or 1) the repeated vectors are assigned the same value with *CurrentVector* and *NextVector*. The MRV procedure outputs a sequence S' of test vectors where the vectors of set *R* have been modified appropriately to reduce the number of transitions at the primary inputs of the CUT.

To evaluate the effectiveness of the MRV procedure we run several experiments on the non-redundant version of the ISCAS'85 benchmark circuits. For each benchmark circuit a compacted test set, achieving complete fault coverage of single stuck-at faults, was used. In a first set of experiments, the graph is constructed as proposed in [4], that is, the weight on each graph edge is equal to the Hamming distance of the corresponding test vector pair. At first we estimated the power dissipation of the sequences derived by the TVO_VR algorithm. We then applied the MRV procedure on these sequences and estimated the power dissipation. Table 1 presents results regarding (a) test set length, (b) the energy dissipation in the circuit which is estimated by the total number of transitions, (c) the average power dissipation given in average number of transitions per test vector and (d) the peak power dissipation which is estimated by the maximum number of transitions that a test vector pair of the sequence activates. The last two columns of the table present the average and peak power reduction percentages achieved by the MRV procedure respectively. Since the test set length remains unaffected by the MRV procedure, the reduction in energy dissipation is equal to the average power dissipation reduction. We have also run a second set of experiments. In this case, the graph is constructed as proposed in [5], that is, the weight on each graph edge is equal to the sum of the *induced activity function* values of the primary inputs of the CUT that have a transition after the application of the corresponding test vector pair. The results obtained are given in Table 2. It is evident from the results of Tables 1 and 2 that we can achieve considerable average power and energy savings using the MRV procedure. Energy and average power reduction percentages vary from 5.5% up to 26.4% depending on the circuit and the specific test set while 17.6% reduction can be achieved on average. Peak power dissipation when using the MRV procedure either remains the same or is slightly improved.

3.2 Root selection and path ordering

According to the TVO_VR algorithm [3], after constructing the MST using Kruskal's algorithm, we arbitrarily select a node as the root and get the inorder traversal of the tree. We have realized that the selection of the root affects significantly the length of the final test sequence and hence the energy dissipation during testing.

Since the MST does not contain specific information about the order of a node's children, we can use any permutation of them and get a new version of the original tree. Furthermore, depending on the node that is considered as the root of the tree, a child of a node in the original MST can become the parent of that node in another tree. It is obvious that an exponential number of different trees can be formed, which makes it computationally infeasible to check. In order to overcome this problem, we have used a

Table 3. The effect of Path Ordering on Test Sequence Length

Circuit	No Path Ordering			Path Ordering		
	Best	Worst	μ	Best	Worst	μ
c432	89	100	96.1	80	90	84.9
c499	101	108	103.9	97	102	99.6
c880	83	95	89.7	77	84	80.5
c1355	150	163	157.7	136	147	141.7
c1908	217	225	222.9	204	216	209.7
c2670	171	181	177.2	163	171	166.8
c3540	398	408	404.6	378	393	384.1
c5315	215	225	222.2	210	216	212.8
c6288	45	51	48.9	38	43	40.6
c7552	304	319	314.9	290	305	298.1

simple heuristic, henceforth called Path Ordering. We construct a new tree by rearranging the children of each node of the original MST in such a way that the longest paths from root to leaf reside at the left hand side of the tree. The idea behind this heuristic is that paths with small differences in their respective lengths will probably have large common parts and therefore the backtracking performed by TVO_VR will be smaller leading to less repeated vectors.

The procedure used for Path Ordering traverses the original MST and stores the paths in descending order according to their respective lengths. The longest path is the starting point for the construction of the new tree and is assumed to be an initial tree called T'. Then, at each step, the longest among the remaining paths is selected and added to T' as follows: if we denote the selected path as $<v_i, \ldots, v_j, v_k, \ldots, v_m>$, where $<v_i, \ldots, v_j>$ is the common subpath that this path has with the T', we add subpath $<v_k, \ldots, v_m>$ to T' by making node v_k the rightmost child of v_j in T'. Obviously T' may contain a path of shorter length located in between two longer paths, since we are not allowed to change the connections of the original tree.

We made the following experiments to evaluate the efficiency of the Path Ordering heuristic. For each benchmark circuit, we constructed the graph as proposed in [4] and run the TVO_VR algorithm n times, each time selecting a different node as the MST root. We computed the average μ of the test sequence length among the n resulting sequences. We also recorded the longest and shortest test sequence derived. We then repeated the same experiment applying the Path Ordering heuristic at each tree resulting from a different root selection. Results are given in Table 3. We observe that the worst solution, in terms of test sequence length, when Path Ordering is applied is, in most cases, equal or slightly better to the best solution achieved without Path Ordering. Therefore we can lead to the conclusion that an arbitrarily chosen root combined with Path Ordering can provide a good solution.

However, if we take a closer look at the description of the TVO_VR algorithm [3, 7] we can make a better selection for the root of the MST. The order of the test vectors in the inorder traversal of the MST has a direct effect on the final test sequence derived by the TVO_VR algorithm. It seems that the best solution in terms of test sequence length can be derived when choosing the tree that has the longest path among all possible trees and applying the Path Ordering heuristic. Therefore, if a more efficient solution is sought, then we need to find the longest possible path in the original MST, create two trees each one having as root one end of the longest path, perform Path Ordering and choose the best result. A means to find the longest path of a graph could be the use of the Bellman Ford algorithm [9] applied to the leafs of the original MST. This approach obviously requires more computational time but provides a better solution. Reduction percentages regarding this case (denoted as Best Path Ordering or BPO) are given in Tables 4 and 5 for the cases where the graph is constructed as proposed in [4] and [5] respectively. These tables present reduction percentages achieved on test set length, energy, average power and peak power dissipation compared to the TVO_VR algorithm [3]. In both cases, the MRV technique was not applied. Results indicate that Best Path Ordering achieves solutions with significantly less number of test vectors. Furthermore, since the variations in average power dissipation are very small (less than 3.7% in all cases), this reduction in test set length subsequently leads to significant reduction in energy dissipation. Energy dissipation reduction up to 22.9% is observed. We thus come to the conclusion that Best Path Ordering can produce test sequences with significantly smaller test set length and energy dissipation without having a tangible effect on the average power dissipation while the peak power dissipation remains unaffected.

Table 4. Reduction Percentages with Best Path Ordering (graph constructed as proposed in [4])

Circuit	Test Length	Energy	Avg. Power	Peak Power
c432	19.2%	19.4%	0.2%	0.0%
c499	4.0%	3.5%	-0.5%	0.0%
c880	13.5%	13.9%	0.4%	0.0%
c1355	16.0%	17.0%	1.2%	0.0%
c1908	8.5%	7.7%	-0.9%	0.0%
c2670	9.4%	8.8%	-0.7%	0.0%
c3540	6.4%	5.4%	-1.1%	0.0%
c5315	6.7%	6.9%	0.2%	0.0%
c6288	20.8%	18.3%	-3.1%	0.0%
c7552	8.8%	6.8%	-2.2%	0.0%

Table 5. Reduction Percentages with Best Path Ordering (graph constructed as proposed in [5])

Circuit	Test Length	Energy	Avg. Power	Peak Power
c432	18.2%	20.0%	2.2%	0.0%
c499	16.4%	16.5%	0.2%	0.0%
c880	7.0%	8.9%	2.0%	0.0%
c1355	18.8%	21.5%	3.3%	0.0%
c1908	11.1%	9.2%	-2.1%	0.0%
c2670	10.3%	10.2%	-1.2%	0.0%
c3540	6.4%	5.8%	-0.7%	0.0%
c5315	11.2%	11.1%	0.0%	0.0%
c6288	20.0%	22.9%	3.7%	0.0%
c7552	8.5%	8.3%	-0.2%	0.0%

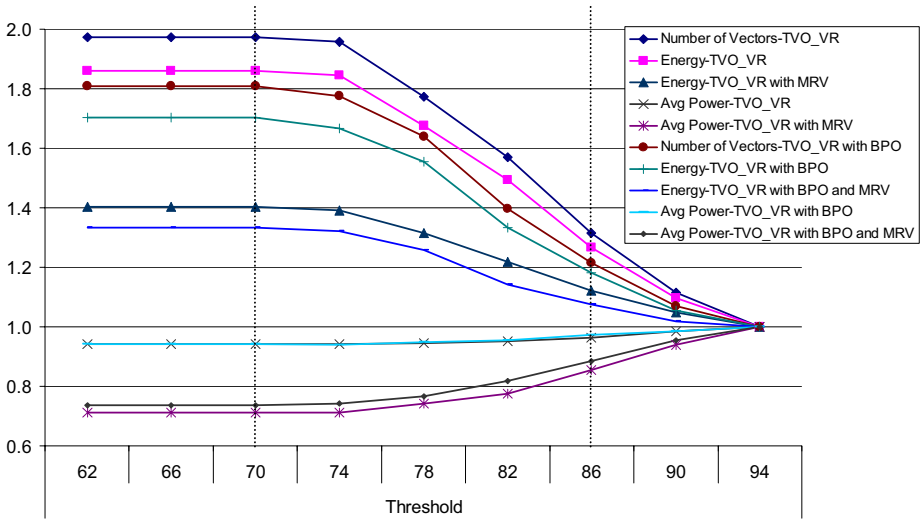

Figure 2. The effect of parameter threshold on c5315 benchmark circuit

3.3 Selecting a value for parameter threshold

There is one more parameter in the TVO_VR algorithm that can be used to control the sequences produced. This is the parameter *threshold*, which is used to decide whether an edge not belonging to the MST should be included to the test vector sequence or not. In [3] this parameter is set equal to the average value of the weights of all MST edges but we can set different values to it. If we set *threshold* to a value smaller than the average value of weights of MST edges, then we expect that the TVO_VR algorithm will add more test vectors in the sequence and will lead to a sequence with more test vectors and larger energy dissipation. However, since edges of minimum weight are inserted to the sequence it will also lead to a solution with less average power dissipation. If, on the contrary, we set parameter *threshold* to a value larger than the average value of weights of MST edges, then more edges will be accepted and therefore a solution with less number of test vectors will be produced. This solution will have smaller energy dissipation at the cost of increased average power dissipation.

We examined the effect of parameter *threshold* on the test sequence length, the energy dissipation and the average power dissipation conducting several experiments. In these experiments the graph is constructed using Hamming distances as proposed in [4]. For each benchmark circuit and compacted test set we run the TVO_VR algorithm using different values for parameter *threshold*. Since results obtained are similar, we present in Figure 2 results only for the c5315 benchmark circuit. The horizontal axis presents the values set to parameter *threshold* while the vertical axis presents normalized values for test sequence length, energy dissipation and average power dissipation. We normalize the results with respect to those of the rightmost value of parameter *threshold*, that is, the value 94. Setting parameter *threshold* equal to 94 denotes the case where the TVO_VR adds no repeated vectors in the test sequence, that is, the length of the produced test sequence has the same length with the original test set. Two dotted vertical lines also exist in Figure 2. The leftmost line indicates the case where *threshold* is set equal to the average value of the weights of all MST edges, while the rightmost line indicates the case where *threshold* is set equal to the average of the weights of all graph edges.

Figure 2 shows that when we set parameter *threshold* to a value less than the one corresponding to the leftmost dotted line, we get the same test sequences. In general, setting parameter *threshold* to a value less than the average value of all MST edges results in test sequences with slightly more vectors and energy dissipation and slightly less average power dissipation. On the other hand, when we increase the value of parameter *threshold* we get test sequences with less number of vectors and energy dissipation and more average power dissipation. Results also reveal that there exists a region where the number of test vectors and the energy dissipation are drastically reduced at the expense of a small increase in average power dissipation. This region is located between the two dotted lines. Therefore we conclude that a good value for parameter *threshold* is between the average of values of all MST edges and the average of values of all graph edges. The diagram also presents results for the case where MRV and/or BPO are used along with the TVO_VR algorithm. As expected, the gains in energy and average power dissipation using the MRV procedure are larger for smaller values of the parameter *threshold* and they decrease as we move to larger values of *threshold* since the test sequences produced in the latter case have less repeated vectors. Furthermore, regardless of the value of the *threshold* parameter, the test sequences derived when applying the BPO heuristic have less number of vectors and less energy dissipation while the average power dissipation remains either the same or is slightly increased.

4. Comparisons

We now present the accumulative effect of the techniques proposed in this paper against the algorithm presented in [3]. In order to obtain a clear picture, we have to use the same values for the parameter *threshold*. For each benchmark circuit we first run the TVO_VR algorithm as described in [3] and then we run the algorithm again applying the Best Path Ordering heuristic on the MST and the MRV procedure on the sequence produced. In both runs, parameter *threshold* was set as in [3].

The reduction percentages achieved by the proposed techniques, in terms of test sequence length, energy dissipation, average power dissipation and peak power dissipation, with respect to the method given in [3] are listed in Tables 6 and 7 for the cases where the graph is constructed in both runs using Hamming distances [4] and primary input's induced activity function values [5] respectively. Results indicate that the proposed techniques achieve significant reductions in test sequence length as well as in energy and average power dissipation whereas the peak power dissipation is practically not influenced. One can also use a larger value for parameter *threshold* and consequently get shorter test sequences with less energy dissipation at the cost of a slight increase in average power dissipation.

Test vector ordering without vector repetition is capable of reducing the average power dissipation during testing up to some point. When further average power reduction is required, we have to resort to test vector ordering with repetition of test vectors [3]. Moreover, in this case, we have the ability to control the average power dissipation by suitably setting the value of parameter *threshold*. Furthermore, the test vector ordering without vector repetition cannot guarantee any reduction in peak power dissipation. On the contrary, the way of deriving the test vector sequence in [3] and in our method guarantees a reduction in peak power dissipation.

Table 6. Reductions achieved by the proposed techniques compared to [3] (graph constructed as proposed in [4])

Circuit	Test Length	Energy	Avg. Power	Peak Power
c432	19.2%	34.4%	17.3%	2.9%
c499	4.0%	12.4%	8.7%	3.4%
c880	13.5%	27.9%	16.7%	0.0%
c1355	16.0%	22.1%	7.2%	2.2%
c1908	8.5%	20.9%	13.5%	0.0%
c2670	9.4%	28.3%	20.8%	0.0%
c3540	6.4%	22.2%	16.9%	-0.1%
c5315	6.7%	27.9%	22.7%	1.1%
c6288	20.8%	26.3%	6.9%	1.2%
c7552	8.8%	25.8%	18.6%	0.0%

Table 7. Reductions achieved by the proposed techniques compared to [3] (graph constructed as proposed in [5])

Circuit	Test Length	Energy	Avg. Power	Peak Power
c432	18.2%	34.1%	19.5%	0.0%
c499	16.4%	20.9%	5.4%	0.0%
c880	7.0%	27.0%	21.5%	0.0%
c1355	18.8%	24.7%	7.3%	1.6%
c1908	11.1%	22.2%	12.5%	0.0%
c2670	10.3%	30.1%	21.2%	0.0%
c3540	6.4%	22.1%	16.7%	0.0%
c5315	11.2%	30.2%	21.4%	1.5%
c6288	20.0%	29.6%	12.0%	0.0%
c7552	8.5%	27.7%	21.0%	0.0%

5. Conclusions

Based on the test vector ordering with vector repetition algorithm originally presented in [3], we presented two techniques, ModifyRepeatedVectors and Best Path Ordering that can be used along with TVO_VR. ModifyRepeatedVectors reduces the switching activity at the CUT's primary inputs by modifying the vectors that appear more than once in the test sequence derived by TVO_VR leading to smaller energy and average power dissipation during testing. BestPathOrdering reduces the length of the resulting test sequence, hence the energy dissipation during testing, by suitably choosing the root and rearranging the nodes of the MST used by the TVO_VR algorithm. Furthermore, we have shown how the value of parameter *threshold* affects average power reduction as well as the test sequence length and energy dissipation during testing.

6. References

[1] P. Girard, "Survey of Low-Power Testing of VLSI Systems", IEEE Design & Test of Computers, pp. 82-92, May-June 2002.

[2] Nicolici, N. and Al-Hashimi, B., *Power-Constrained Testing of VLSI Circuits*, Kluwer Academic Publishers, 2003.

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

[4] P. Girard, C. Landrault, S. Pravossoudovitch and D. Severac, "Reducing Power Consumption during Test Application by Test Vector Ordering", Proc. of International Symposium on Circuits and Systems, pp. 296-299, 1998.

[5] P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, "A Test Vector Ordering Technique for Switching Activity Reduction during Test Operation", Proc. of 9th Great Lakes Symposium on VLSI, pp. 24-27, 1999.

[6] Z. Luo, X. Li, H. Li, S. Yang and Y. Min, "Test Power Optimization Techniques for CMOS Circuits", Proc. of Asian Test Symposium, pp. 332-337, 2002.

[7] V. Dabholkar and S. Chakravarty, "Minimizing Power Dissipation in Combinational Circuits During Test Application", Technical Report 94-10, Dept. of Computer Science and Engineering, University at Buffalo, 1994.

[8] A. Shen, A. Ghosh, S. Devadas and K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks", Proc. of International Conference on CAD, pp. 402-407, 1992.

[9] Cormen, T. H., Leiserson, C. E. and R. L. Rivest, *Introduction to Algorithms*, MIT Press, 1990.

0-7695-2093-6/04 $20.00 © 2004 IEEE