International Journal of Engineering & Technology, 7 (4) (2018) 4357-4361

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10498
Research paper



Enhanced compaction and reordering procedure for transition fault testing

Navya Mohan [1] *, J. P. Anita [1]

[1] *Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore. Amrita Vishwa Vidyapeetham, India*
**Corresponding author E-mail: m_navya@cb.amrita.edu*

Abstract

Compaction procedure, used to reduce test power, can be efficiently applied on test cubes or incompletely specified tests. This paper demonstrates an improved static compaction and switching activity based test vector reordering methodology which can be applied on a test set that contains both skewed load and broadside tests for transition faults. This compaction procedure goes beyond the normal test vector merging approach generally employed in testing circuits. Here a test is combined with several other tests even if they are not compatible. After obtaining a compact test set, the vectors are reordered such that the total switching activity of the circuit including all the internal nodes is reduced. The simulation results show a considerable reduction in the number of tests and as the test volume reduces the test power also decreases. An average 28.6% reduction in number of test vector pairs is observed as compared to existing static compaction methods and 30.6% reduction in average switching activity (ASA) after reordering.

Keywords: *Average Switching Activity; Broadside Tests; Reordering; Skewed-Load Tests; Static Test Compaction; Test Cubes; Transition Faults.*

1. Introduction

In recent years highly complicated integrated circuits are being designed and developed where the number of logic gates and components is also increasing exponentially. All this advancement has occurred due to the development in the semiconductor industry. As the number of components increases, the complexity of the circuits also increases. Thus while testing complex circuits simple stuck-at fault models are no longer sufficient. It has become very important to consider other types of faults like transition faults while testing a circuit. If a transition at node N does not arrive either at the scan flip-flop or a primary output within the predetermined clock period of the circuit then it is considered as a transition fault.

On any given line in a circuit there can be two kinds of transition faults namely Slow-to-Rise transition fault (STR) and Slow-to-Fall (STF) transition fault. When a circuit is operating at its maximum operating frequency and a transition from logical level 0 to 1 at a node does not give rise to required output then it is said to have STR fault at that particular node. Similarly, if a transition from logical level 1 to 0 on a node does not result in the correct response at full operating frequency then it is called as STF fault. In order to identify a transition fault successfully, a single test vector is insufficient. A test vector pair $v = \{t_1, t_2\}$ has to be applied successively, where t_1 is the set of input values that fixes the target node to a required preliminary value and the next input vector t_2 should be selected in such a way that it is capable of launching the required transition at the node, as well as propagating the effect of the transition to at least one primary output or a scan flip-flop.

According to literature there are two ways in which transition faults can be detected. The first method is broad-side delay test which can also be called as Launch-on-Capture technique (LOC) [1]-[3]. In this method, the first test vector among the pair is loaded into the scan chain and the subsequent test vector is obtained as the response of the combinational part of the circuit to the first vector. The second method is skewed-load transition testing which is also called as the Launch-On-Shift technique (LOS) [1] and [3]. In this method, both the vectors in a pair are given as input through the scan chain. It takes N clock cycles to load the bits if length of the scan chain is N-bits. The first (N-1) bits are scanned in (N-1) clock cycles and the N^{th} clock pulse is used to commence the transition, followed by an immediate capture.

The starting test set used in this paper for application of static compaction consists of both skewed-load as well as broadside test. It is proven in [1] that this type of test is advantageous in improving the transition fault coverage. Hence the test compaction method used here is aimed at reducing the test volume while maintaining the fault coverage. This is possible by generating a test set in which a test detects as many faults as possible. There are two ways of compaction – static and dynamic. Dynamic compaction gives a reduced number of test vectors at the test generation stage itself. However static compaction procedure first generates all the tests and then merges the compatible test vectors. Both static and dynamic test compaction methods make use of the unspecified bits in the test set [4], [5] and [8].

For a transition fault, a two pattern test is always considered because to detect transition fault at a node, a set of two test vectors are needed. The proposed algorithm can also be applied for stuck-at-fault compaction in addition to transition fault. In addition to reducing the test volume while preserving the fault coverage, a test set reordering methodology to reduce the test power is also discussed. Many test vector reordering methods have been discussed in the literature. A reordering method based on hamming distance is discussed in [13] and a reordering method for improving diagnostic resolution is mentioned in [13] and [14]. The reordering method proposes a graph based heuristic approach for reducing the total number of transition occurring in a circuit during test.

Copyright © 2018 Navya Mohan, J. P. Anita. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is organized in four sections. Section 1 gives an introduction and overview of the existing work in this area, Section 2 discusses the proposed method in detail, Section 3 illustrates the results and comparisons with some existing compaction methods and finally Section 4 gives a brief conclusion and future scope of the work.

2. Proposed method

In this section an improved compaction procedure is presented in detail. This is rooted on the idea of combining a test pair with different possible test pairs even if they are not of the same type.

Method of generating a combined test set (containing both broadside and skewed load) is discussed in [11] and [13]. In a normal test compaction procedure using test merging, two test pairs v_i and v_j are merged to get a combined test pair v_{i+j} if they have no conflicting bit positions. The resultant test pair v_{i+j} will identify all the transition faults uncovered by v_i and v_j. In addition to that it may detect some additional faults due to the extra specified bits. This may decrease the percentage of relaxation but is highly useful in reducing the test volume.

For testing a sequential circuit in addition to the primary input bits the state of the flip-flops should also be specified. Hence any test vector will consist of two parts namely scan-in bits (S_i) and primary input bits (P_i). Hence for detecting a transition fault we will need a test pair which can be represented as $V_i = (S_{i1}\ P_{i1}, S_{i2}\ P_{i2})$.

This work presents a test compaction method where instead of simply merging the compatible test vectors, it tries to combine tests which may not be compatible and may be of different type. Two test pairs are said to be of different type if one is broadside test pattern and the other is skewed load pattern. Two tests are said to be compatible if they have no conflicting '1' or '0' in any position. Two test vector pairs $v_{i1} = \{ S_{i1,1}\ P_{i1,1}, S_{i2,1}\ P_{i2,1}\}$ and $v_{i2} = \{ S_{i1,2}\ P_{i1,2}, S_{i2,2}\ P_{i2,2}\}$ are said to be compatible if the following conditions are fulfilled:

a) Both are skewed-load or both are broadside i.e. they are of same type.
b) Primary input vectors are compatible, i.e. $P_{i1,1}$ is compatible with $P_{i1,2}$ and $P_{i2,1}$ is compatible with $P_{i2,2}$.
c) Scan in states are compatible i.e. $S_{i1,1}$ is compatible with $S_{i1,2}$ and $S_{i2,1}$ is compatible with $S_{i2,2}$.

For instance consider the benchmark circuit s27from ISCAS' 89 shown in Fig.1. Here the nets G0 to G3 are the primary inputs P_i and G13, G10, G11 are the scan inputs S_i.

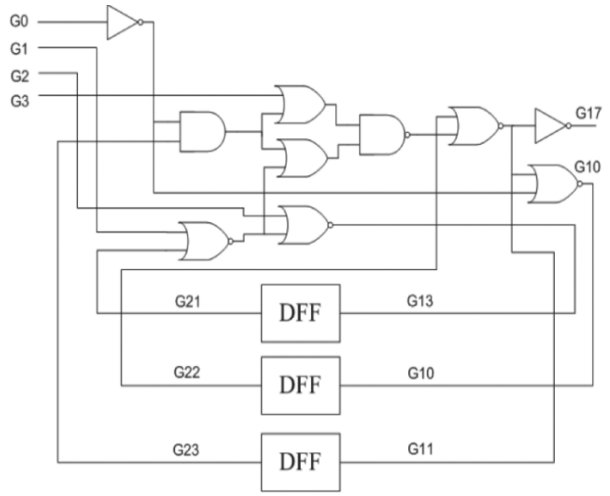


Fig. 1: S27 Benchmark Circuit.

Table 1 shows a set of test vector pairs for s27 benchmark circuit. Consider test pairs v_0 and v_1, even though they satisfy condition (a) and (b) of compatibility the condition (c) is not being satisfied. Hence they are not compatible. But the test pairs for i=1 and 2 (i.e. v_1 and v_2) satisfy all the three conditions mentioned for compatibility, so they are considered as compatible test pairs. Thus vectors v_1 and v_2 can be replaced by a single test pair v_{1+2} which will uncover all the faults identified by v_1 and v_2. Table 2. shows the set of test pairs after applying a static compaction method on the set of test pairs for the circuit s27 shown in Table 1. Here the following tests are compatible as they satisfy all the three compatibility conditions:

1) v_0 and v_7
2) v_1 and v_2
3) v_8 and v_9

The compact test set obtained after applying test merging procedure is shown in Table 2. Here v_7, v_2 and v_9 are removed and v_0, v_1 and v_8 are replaced by v_{0+7}, v_{1+2} and v_{8+9}

Table 1: Sample Test Set for S27

Test No. (v_i)	Type	S_{i1}	P_{i1}	S_{i2}	P_{i2}
v_0	Brd	0x0	x0x1	010	11x1
v_1	Brd	01x	1xx0	01x	0xxx
v_2	Brd	xx1	xx0x	xx1	xx1x
v_3	Skw	10x	1xxx	010	0xx0
v_4	Brd	111	0x0x	0x1	0xx1
v_5	Skw	11x	0xxx	011	0xxx
v_6	Brd	x0x	0x10	000	x0x1
v_7	Brd	xxx	x01x	xx0	x10x
v_8	Brd	xx1	xx0x	xx1	xx1x
v_9	Brd	1xx	1xxx	10x	0xxx

Table 2: Sample Test Set after Test Merging

Test No. (v_i)	Type	S_{i1}	P_{i1}	S_{i2}	P_{i2}
v_{0+7}	Brd	0x0	x011	010	1101
v_{1+2}	Brd	011	1x00	011	0x1x
v_3	Skw	10x	1xxx	010	0xx0
v_4	Brd	111	0x0x	0x1	0xx1
v_5	Skw	11x	0xxx	011	0xxx
v_6	Brd	x0x	0x10	000	x0x1
v_{8+9}	Brd	1x1	1x0x	101	0x1x

In this approach the test pairs v4 and v5 cannot be merged as they are of different types. This drawback is rectified in the proposed method where they can be replaced by a single test pair v_5' in which the type and conflicting bit positions are assigned the values of the first test pair under consideration. In the above illustration v4 is the first pair. Thus the resulting test pair v_5' will be {111 0x0x, 011 0xx1} and of the type 'Brd'. v_5' will be able to uncover all the faults identified by v_4 and some faults identified by v_5.

Table 3 shows four different scenarios in which the normal test merging procedure would fail. First example is that of combining two broadside test pairs. Combined test pair i_1+i_2 is also of the type broadside. Moreover if there is any conflicting bit position, then it is assigned with the corresponding bit value in i_1. Third example is of two test pairs which are of different types. Here the combined test set i_1+i_2 will be of the type skewed-load same as that of i_1.

Table 3: Example of Combining Incompatible Test Pairs

Test	Type	S_{i1}	P_{i1}	S_{i2}	P_{i2}
i1	Brd	0x0	x0x1	010	11x1
i2	Brd	01x	1xx0	01x	0xxx
i1+i2	Brd	010	10x1	010	11x1
i1	Skw	10x	1xxx	010	0xx0
i2	Skw	11x	0xxx	011	0xxx
i1+i2	Skw	11x	1xxx	010	0xx0
i1	Brd	111	0x0x	0x1	0xx1
i2	Skw	11x	0xxx	011	0xxx
i1+i2	Brd	111	0x0x	011	0xx1
i1	Skw	10x	1xxx	010	0xx0
i2	Brd	111	0x0x	0x1	0xx1
i1+i2	Skw	101	1x0x	010	0xx0

2.1. Algorithm

Input to this combination-based algorithm is a set of incompletely specified test vector pairs $V = \{v_0, v_1, \ldots, v_{m-1}\}$ and a transition fault list F. This is an iterative algorithm where one test pair is removed per iteration. This algorithm does not follow a conventional test vector merging approach. Instead of merging v_{i2} completely with

another test it follows an approach where v_{i2} is merged with different test pairs. Initially, it merges v_{i2} with every other test pair in V. If after the merging process, the resulting test set pair detects all the transition faults which were detected by v_{i2} then v_{i2} can be removed. The algorithm starts with a reverse order fault simulation including fault dropping of F under V. For each test vector pair $v_i \in$ V, it generates a fault set $E_i \subseteq$ F that are identified by v_i. Since fault-dropping method is incorporated, the test vectors that appear later in V have smaller set of detected faults as compared to the vectors considered earlier. If V= {v_0, v_1,..,v_n}, then they are considered in the order $v_{n,}$ v_{n-1},...,v_0. This order is chosen because the set of detected faults by test vectors present at the end is smaller, so removing such test vectors is easier [9].

At first V is copied into a test pair denoted by V_{pre} so that the initial test set can be recovered if needed. It is then verified whether vectors in V other than v_{i2} detect some of the transition faults in the set E_{i2}. This is achieved with the help of fault simulation including fault dropping of E_{i2} under V − {v_{i2}}. After updating E_{i2}, if $E_{i2} = \emptyset$, then the test v_{i2} can be removed without reducing the fault coverage. On the other hand if $E_{i2} \neq \emptyset$, every test $v_{i1} \in$V is replaced with the test v_{i1+i2} obtained by merging v_{i1} with v_{i2}.Then fault simulation including fault dropping of E_{i2} under V − {v_{i2}}is performed. Consider a fault f ∈ E_{i2}, if it is detected by a test vector $v_{i1} \in$ V − {v_{i2}}, move f from E_{i2} to E_{i1} and include the index 'i_1' in a set I_1. If $E_{i2} \neq \emptyset$, then v_{i2} cannot be removed so, recover the earlier test set by fixing V = V_{pre} and need not take into account v_{i2} here after.

If a test vector $v_{i1} \in$ V such that $i_1 \in I_1$ is not helpful in identifying faults from E_{i2} or any fresh faults then the combined test vector v_{i1+i2} is of no use. Thus every test $v_{i1} \in$ V such that i1 ∉ I_1 can be replaced with $v_{i1} \in V_{pre}$. Hence it is safe to remove test v_{i2} without compromising on transition fault coverage.

An improved test compaction procedure was first proposed in [1]. In this method, after the merging procedure is over, every specified bit in the resultant test set is considered for possible relaxation without loss in fault detection. This step however adds to the run time of the algorithm. Thus to compensate for the increase in run time the procedure stops before considering all the tests for possible removal, which results in increased number of test vectors. Due to the above considerations the test size obtained in [1] is not optimal. To resolve this problem in the proposed method a lower bound of 20% is set for the unspecified bits in the final test set and the procedure is run until all the tests are considered for removal. This guarantees a highly optimized test set.

3. Test pair reordering

After obtaining the compact and relaxed test set, test reordering algorithm can now be applied. Since for detecting a set of transition faults, test vector pairs are needed, the order of each test pair cannot be altered while applying the reordering algorithm. Test vector reordering method reported in the literature employs a Hamming distance based approach. This method assumes a high correlation between the transition density at the primary input of Circuit Under Test (CUT) and switching activity in the circuit. However, this supposition is not always true, because a single transition at a primary input node may trigger several transitions in the intermediate nodes, whereas change in several inputs may cause fewer transitions in the internal nodes [10] and [12]. In order to obtain a considerably better vector reordering, the transitions at all the internal nodes have to be considered.

Consider a test vector pair v_i= (t_i , s_i) where t_i is the first vector and si is the second vector. Let the initial order of the test vector pair be v1, v2, v3,..,vi,vi+1,..vn. A test vector t_i is applied to the circuit followed by s_i, this order cannot be changed as they form a test vector pair used to detect one or more transition faults. A complete undirected graph G = (V,E) is constructed in which each vertex $V_1 \in$ V represents a test pair and each undirected edge $E_{i,j}$ = (V_i, V_j) ∈ E connects two test pairs. The weight on each edge $E_{i,j}$ represents the cost in terms of switching activity for the application of test sequence V_i,V_j to the circuit. Weight on edge E_{ij} can be defined as the number of nodes that are switching in the CUT when V_j is applied after V_i. While calculating the weights on the edges, the unspecified bits can be exploited to reduce the transition activity. A Minimum Transition (MT)-filling method can be used to reduce switching activity [4]. MT-filling is not a necessary condition for applying reordering procedure but by the application of it the test power can be reduced even further.

4. Results and analysis

Transition faults were injected into the benchmark circuits using Synopsys TetraMAX fault simulator. Initial set of test vectors used to detect these faults were generated using Synopsys-TetraMAX ATPG. Power dissipation before and after compaction was computed using Synopsys-PrimeTime. The tool was run on a HP computer with Intel® Core™-i5-2500 processor, 3.30 GHz clock and 8GB RAM. The compaction algorithm, reordering algorithm and procedure for calculating the total number of transitions was implemented in C language.

This procedure follows a static compaction methodology and is applied to different ISCAS-89 benchmark circuits as shown in the Table 4. The second column in Table 4 shows the initial set of test vector pairs obtained from TETRAMAX ATPG and the third column indicates the number of tests after applying a normal test vector merging approach and the last column shows the reduced number of test vector pairs obtained after applying the proposed compaction algorithm while maintaining the fault coverage. Even though a simple test merging approach gives reasonably compact test vectors for stuck-at faults; it fails to do so for transition faults. From the results obtained in Table 4 it can be observed that normal test vector merging approach does not provide highly compact test set, due to its inherent limitation where test pairs of same type (broadside or skewed-load) can only be combined. However this limitation is removed in the proposed method, thus providing highly compacted test set without compromising fault coverage.

Table 5 compares the number of test pairs obtained after applying the proposed compaction algorithm with that of a compaction algorithm for transition fault mentioned in [1]. It can be inferred that the proposed method is giving lesser number of tests while maintaining the fault coverage. This improved result is obtained at the cost of relaxed bits. But due to the decrease in the test size and application of reordering algorithm the power dissipated during testing can be reduced to a greater extent.

Table 4:.Number of Test Pairs before and after Compaction

Circuit	No. test vector pairs before compaction	No. test vector pairs Using normal Test Merging	No. of test vector pairs in proposed method
s27	24	17	9
s208	135	77	30
s298	275	114	51
s344	387	298	31
s349	432	261	25
s382	504	435	35
s386	890	561	60
s400	650	326	22
s444	370	235	17
s526	843	432	46

As the number of tests decreases the power consumed during testing of the circuit also decreases [6], [7] and [9]. This has been validated by the results shown in Table 6. Switching power increases when signals change their logic state. The dynamic power of the circuit is its internal power and switching power together. Leakage power is the power consumed by the circuit in idle state. It is not related to the switching activity of the circuit. Internal power mainly constitutes the power dissipated during the brief short between the supply and ground due to the switching. The power dissipated at the output of a cell during charging and discharging of the capacitance is the switching power.

Table 5: Comparison of Number of Test Pairs obtained by The Proposed Method with [1]

Circuit	[1]	Proposed	% Reduction
s27	13	9	30
s526	70	46	34.2
s1196	158	101	36.07
s5378	213	131	38.4
s9234	301	245	18.6
s13207	419	345	17.6
s15850	235	175	25.5

The switching power values given in Table 6 are net values that depend on the transitions of input bits fed to the circuit. The third and seventh column in Table 6 shows the internal power values of different sequential benchmark circuits. Internal power is the power consumed when the input given to the gate changes but the output doesn't change. In logic gates not every change input cell necessarily leads to a change in the state of the output net. As the complexity of the circuit increases then the internal power of the circuit increases.

Fourth column in Table 6 shows the leakage power values of different sequential benchmark circuits. The leakage power is observed to be high for all the circuits. As the complexity of the circuit increases the leakage power also tends to increase. The more the complexity of the circuit, more will be the area required and more will be the power consumed.

Table 6: Total Power before and after Compaction

	Before compaction				After compaction			
Circuit	Switching Power (mW)	Internal power (mW)	Leakage power (mW)	Total power (mW)	Switching Power (mW)	Internal power (mW)	Leakage power (mW)	Total power (mW)
s27	0.04519	0.1383	0.3319	0.5154	0.004508	0.01025	0.3212	0.3360
s208	0.04628	0.1263	0.3245	0.5296	0.004567	0.01074	0.3224	0.3254
s298	0.0006836	0.8802	3.725	4.606	0.0002631	0.3240	3.557	3.882
s344	0.6255	0.7805	4.603	6.009	0.1614	0.2135	4.619	4.994
s349	0.6532	0.7773	4.632	6.603	0.1407	0.1456	4.830	4.830
s382	0.0007290	1.094	3.570	4.665	0.0001890	0.2766	3.654	3.931
s386	0.00008789	0.1307	7.384	7.514	0.00006143	0.06409	7.326	7.390
s400	0.0007703	1.183	3.684	4.868	0.0001181	0.1865	3.759	3.946
s444	0.0007153	1.0434	3.449	4.449	0.00007974	0.1223	3.374	3.496
s526	0.0008705	1.256	6.094	7.351	0.0004115	0.5854	6.021	6.607

Table 7 describes the percentage reduction in net switching power, internal power, leakage power and also the total power. There is a maximum decrease of 38.55% of total power for s208 and minimum decrease of 1.65% for s386 because the switching power also depends on the order in which the input test vectors are applied. However, for all the circuits considered above, it can be observed that the power dissipation has considerably reduced after applying the test compaction algorithm. As the leakage power is not dependent on the switching activity, there is no considerable reduction in it after compaction.

Table 7: Percentage Reduction in Power before and after Compaction

Circuit	Decrease in Switching Power (%)	Decrease in Internal Power (%)	Decrease in Leakage Power (%)	Decrease in Total Power (%)
s27	90.02	92.59	3.22	34.81
s208	90.13	91.50	0.64	38.55
s298	61.51	63.19	4.51	15.71
s344	74.20	72.65	0.34	16.89
s349	78.46	81.26	1.92	26.85
s382	74.07	74.72	2.35	15.73
s386	30.11	50.96	0.78	1.65
s400	92.77	84.23	2.03	18.94
5444	88.85	88.28	2.17	21.42
5526	52.73	53.39	1.19	10.12

After performing test compaction, test pairs are reordered to reduce the power even further. Table 8 shows the comparison of the total number of transition occurring in each benchmark circuit before and after reordering. It can be observed that the number of transition is considerably reduced after reordering. Transition occurring in a circuit is directly related to the switching power in a circuit [4]. As the transition decreases, the power dissipated during testing also decreases. Hence a separate power analysis is not performed.

Table 8: Total Number of Transitions before and after Reordering

circuit	No. test vectors	Before reordering	After reordering
s27	9	37	22
s298	51	2364	1480
s382	35	4681	3900
s386	60	13738	10642
s526	46	4570	3902
s1196	158	27353	17857
s5378	213	46236	24342

Average switching activity (ASA) is the number of transitions per test vector. It is the ratio of total number of transitions by total number of test vectors. As compared to the total number of transition, ASA is superior unit for assessment of results as it is directly proportional to the power consumed during testing. It implies that if the switching activity is less, then the test power is also less [4].

Table 9 shows the average switching activity for ISCAS'89 benchmark circuits before and after reordering. It can be observed that ASA has reduced by a maximum of 47.3% for s5378 and minimum of 14.6% for s526.

Table 9: Average-Switching Activity before and after Reordering

Circuit	No. test vectors	before reordering	after reordering	% change
s27	9	4.1	2.4	41.4
s298	51	46.3	29.0	37.6
s382	35	133.7	111.4	16.3
s386	60	228.9	177.4	22.5
s526	46	99.34	84.8	14.6
s1196	158	173.1	113.0	34.6
s5378	213	217	114.3	47.3

5. Conclusion and future scope

Transition faults occur in digital circuits because of propagation delay in interconnects and finite rise and fall times of signals. Today's semiconductor technology is increasing its emphasis on at-speed testing. Transition faults play a very important role here as it helps to screen out delay defects. In the proposed method, the quality of test merging is enhanced by combining tests which are of different types.

The main factors considered while testing a given circuit is testing time and power. As the test pattern count decreases, the time consumed during testing also decreases. The simulation results obtained have proved that there is a substantial reduction in number of tests and hence the power consumed while testing also reduces. By applying test reordering, the ASA has been reduced noticeably. The test application time can further be reduced by applying test compression techniques.

References

[1] Pomeranz, I. (2013). On test compaction of broadside and skewed-load test cubes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *21*(9), 1705-1714. https://doi.org/10.1109/TVLSI.2012.2217360.

[2] Ang, C. H. (2013, November). Single Test Clock with Programmable Clock Enable Constraints for Multi-clock Domain SoC ATPG Testing. In *Test Symposium (ATS), 2013 22nd Asian* (pp. 195-200). IEEE.

[3] Saeed, S. M., & Sinanoglu, O. (2014). Design for testability support for launch and capture power reduction in launch-off-shift and launch-off-capture testing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *22*(3), 516-521. https://doi.org/10.1109/TVLSI.2013.2248764.

[4] Mohan, N., & Anita, J. P. (2016). A zero suppressed binary decision diagram-based test set relaxation for single and multiple stuck-at faults. *International Journal of Mathematical Modelling and Numerical Optimisation*, *7*(1), 83-96. https://doi.org/10.1504/IJMMNO.2016.074374.

[5] Pomeranz, I. (2011, May). Static test compaction for delay fault test sets consisting of broadside and skewed-load tests. In *VLSI Test Symposium (VTS), 2011 IEEE 29th* (pp. 84-89). IEEE. https://doi.org/10.1109/VTS.2011.5783760.

[6] Pomeranz, I., & Reddy, S. M. (2011). Static Test Data Volume Reduction Using Complementation or Modulo-$ M $ Addition. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *19*(6), 1108-1112. https://doi.org/10.1109/TVLSI.2010.2044819.

[7] Mohan, N., Krishnan, M., Rai, S. K., MathuMeitha, M., & Sivakalyan, S. (2017, September). Efficient test scheduling for reusable BIST in 3D stacked ICs. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on* (pp. 1349-1355). IEEE.

[8] Pomeranz, I. (2016). Static test compaction for circuits with multiple independent scan chains. *IET Computers & Digital Techniques*, *10*(1), 12-17. https://doi.org/10.1049/iet-cdt.2014.0191.

[9] Naeini, M. M., Dass, S. B., Ooi, C. Y., Yoneda, T., & Inoue, M. (2017). An integrated DFT solution for power reduction in scan test applications by low power gating scan cell. *Integration, the VLSI Journal*, *57*, 108-124. https://doi.org/10.1016/j.vlsi.2016.12.009.